Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Experimental analysis of indoor location systems for smart devices

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Emilio José Pérez Salgado

DIRECTOR: Davide Vega

DATE: November 1st, 2014

TITLE: Experimental analysis of indoor location systems for smart devices

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Emilio José Pérez Salgado

DIRECTOR: Davide Vega

Data: November 1st, 2014

Resumen

Desde finales de los 90, con la incorporación de las PDA y las cajas registradoras táctiles para tomar nota de los pedidos, no ha habido un cambio significativo en cuanto al uso de la tecnología en el sector de la restauración. Actualmente las aplicaciones para dispositivos móviles se están introduciendo en este sector. Hoy en día ya existen webs y aplicaciones que ofrecen ofertas y reservas y, en estos últimos años, también han aparecido aplicaciones que permiten hacer pedidos online para comida a domicilio.

El siguiente salto tecnológico en el sector de la restauración será tener un Smart-Restaurant en el que un individuo a través de su dispositivo móvil pueda obtener información del aforo del restaurante, hacer una reserva, descargar la carta, seleccionar lo que quiere comer modificándolo a su gusto y, por ultimo, pagar. Todo ello controlado a tiempo real y con el fin de optimizar los tiempos de servicio. Para ello es importante tener localizada a la persona dentro del recinto, con el fin de controlar los pedidos y la gestión del servicio en general.

Este estudio pretende analizar la parte relacionada con el posicionamiento del usuario a lo largo de todo el proceso; desde que selecciona un restaurante hasta que lo abandona. Para ello se utilizarán herramientas de posicionamiento en exteriores e interiores, aunque este documento se centra principalmente en el posicionamiento en interiores.

El análisis de dichas herramientas se centrará en ver la capacidad que tienen para poder situar a un individuo dentro de un restaurante, en una mesa concreta y en una silla en concreto, con el fin de poder desarrollar un sistema de gestión inteligente.

TITLE: Experimental analysis of indoor location systems for smart devices

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Emilio José Pérez Salgado

DIRECTOR: Davide Vega

Data: November 1st, 2014

Overview

Since the late 90's with the introduction of the PDA and cash register to take orders, there has not been a significant change in the use of technology in the catering industry. Nowadays, new technologies are currently being introduced in this industry. There are websites and applications that offer deals and reservations, and, in the last years have appeared applications that allow online orders to take away.

The next step in the catering sector will be a Smart-Restaurant, in which individuals, through their mobile devices, could obtain information about gauging of the restaurant, make a reservation, download the menu, select what they want to eat, customizing it to their liking and, finally, pay. All this would be controlled in real time to optimize the time of service. For that purpose, it is important to keep the person located inside the restaurant, in order to control and manage the services requested.

This Master Thesis analyzes the section regarding to the positioning of the user through the entire process; from selecting a restaurant until he leaves. For that purpose, I will use positioning tools in outdoor and indoor, although this document will primarily focus on indoor positioning.

To develop an intelligent management system, the analysis of the aforementioned tools will be focused on the ability to locate an individual in a specific restaurant, at a specific table and in a specific seat.

A mi hija Sofía, que ha sido fuerza y motivación durante todo el camino.
A mi mujer por su apoyo en esta aventura.
A mi madre y a mi padre por su paciencia y su gran ayuda.
A mi hermano por su inspiración.
A todos aquellos que han confiado en mí.
GRACIAS

"Uno puede devolver un préstamo de dinero, pero está en deuda de por vida con aquellos que le ayudan."
(proverbio)

CONTENT

FIGURE LIST

TABLE LIST

INTRODUCTION

Qpides! is a system consisting of (i) an application and (ii) a server, with the purpose to facilitate the temporal-reservation, selection and acquisition of products/services. In particular, for the catering industry, a customer of a Qpides! could search restaurants in the area and check their menus and availability before actually approaching them.

Once a restaurant is selected, the customer will be automatically assigned to a table for a period of time. Qpides! will guide the customer to the restaurant and then the customer may sit and place his order. Those orders by the customer will be sent to the kitchen using the same device used for searching the restaurant and checking the menu. The whole transaction, linked to customer and table, will be considered as finished when the restaurant's client leaves the restaurant with the bill paid.

For the aforementioned transactions, the location plays a very important role. Qpides! is present in the time to look for a restaurant, when Qpides! guides the client to the selected restaurant and, finally, when Qpides! tells the client where to sit. In all this process, the most critical part is the indoor-location because at the end, it will help to establish whether the client was in the local and determine when the customer will pay the restaurant. When the customer wants to pay the bill, Qpides! will process all the user information (from the selection of the restaurant until the user leaves it). This information is stored on the Qpides! server as a probe that this user on a specific day at a specific time was there.

Motivation

My main motivation was to deploy an indoor location system using Wi-Fi signals because if this was possible, the cost of implementation of Qpides! App in a restaurant would be lower since no external devices would be required.

The Wi-Fi is the most usage signal for interchange information. Therefore, this Thesis is focused on obtain a solution for indoor-location using WI-FI signal.

Also, as a secondary motivation was the fact that this project was experimental rather than theorist, collecting data and analyzing them. So, this thesis also aims to analyze the existing indoor positioning methods based on Wi-Fi signals, and verifies if they have the accuracy requirements necessary to locate individuals within establishments, always from a practical perspective.

This project gives me the chance of programming tools that Google provides with the challenge of learning how to program an App for IOS, and learn about positioning algorithm.

Objectives

This Thesis is based on the capability of Wi-Fi signals to obtain an accurate indoor-location based on the needs of the Qpides! application.

This Thesis will be review different technologies related with indoor-location. Once selected the technology, will be tested using a real space.

This Thesis provides an error margin and uses different user-device to test the results.
The work will conclude with future improvements if the result is satisfactory.

At the end, analysis will develop the ability to locate a person using Wi-Fi signals in a restaurant, at a table in a concrete and specific seat in order to develop an intelligent management system for restaurants.

Document organization.

The project, motivation and objectives are presented in the Introduction chapter.

The first chapter presents Qpides! giving a broad vision on the mobile-application.

The second chapter presents the state of art of the fields associated with this project, tracking systems, the points to be considered for selection and systems/algorithms used for each system.

The third chapter contains the scenario, where presents the mapping area and how obtain the Wi-Fi measurement.

The fourth chapter is focused on the fingerprint and in the results obtained for different systems using the same scenario. In this chapter we can find the selection of the most suitable fingerprint options for our purpose.

The chapter fifth presents the analysis and the evaluation. This section presents the different ways to analyze the data results using Weka software.

The sixth chapter presents the environmental impact.

Finally, the conclusions are in the seventh chapter as a possible continuation of the work or improvements made.

CHAPTER 1. Qpides!

Nowadays, the customers of any product and services are prone to exploit the social intelligence before actually purchasing anything. The current trend, followed by the majority of potential clients, is to search, compare and, possibly, buy merchandise online. They are aware of the vast amount of information available on Internet and how to get most of it.

The later searches are capable to use geo-located information, and therefore become an excellent opportunity to approach the client. This approach would provide him/her with real-time information and lead their steps towards the adequate store.

It is vital to give the possible customer enough and reliable data to satisfy its expectations when visiting the store. Such satisfactory transaction will increase his/her confidence in the local business and increase the area of influence among his/her social group. Thus, new ways of advertisement and customer attraction will appear in the market.

This is then the opportunity of Qpides! to provide a satisfactory platform to search geo-localized retailers willing to fulfill their potential customer needs.

The service that Qpides! offers, it does not stop when the client arrives to the store, but rather it goes with him until his departure of the store, cashing-out his consumption/shopping.

1.1. The idea

Qpides! is based on an infrastructure aimed to facilitate “temporal” reservation, selection and acquisition of products/services. So, Qpides! is present since the client´s wish until the product is finally delivered. In the case of the restaurant’s client, it is expected to search nearby restaurants and check their menus and availability before actually approaching them.

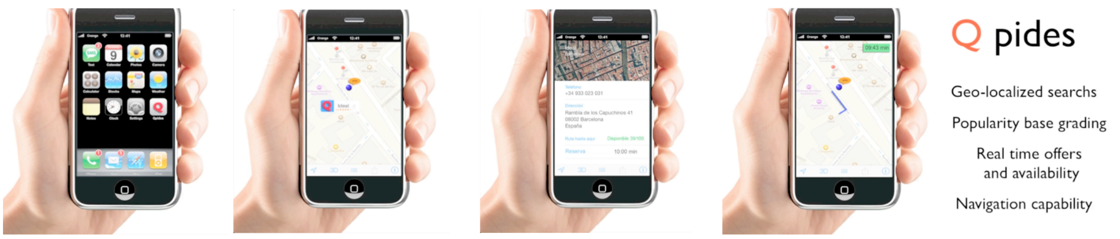


Figure 1. Qpides! User outdoor functionalities

Once a restaurant is selected, the customer will be automatically assigned to a table in a time window, and then he may sit and place his order. Those orders by the customer will be sent to the kitchen using the same device used for searching the restaurant and check the menu. The whole transaction, linked to customer and table, will be considered as finished when the restaurant’s client leaves the restaurant with the bill paid. The last action is also available through agreements with different financial enterprises.

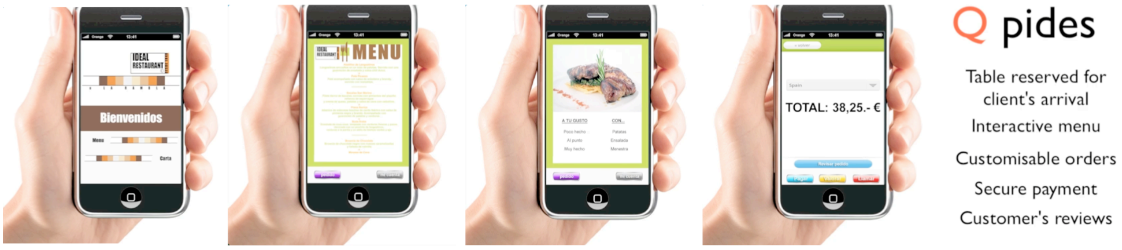


Figure 2. Qpides! User restaurant functionalities

The versatility of the application provides the means to adapt it to any business model where the consumer is required to book, select products or services and fulfill payment.

1.2. Allocation process

The positioning system works with three servers. The Google Maps server locates the user's position outdoor and guides the user to the restaurant.

The Qpides! server, stores information for each restaurant. This part of the project will work with address information (GPS positions) of every restaurant that has the Qpides! system and information from radio signals that characterize each restaurant (SSIDs, Bluetooth-name, Cells-coverage).

Finally the Indoor server works to the user's location inside the restaurant.
The detailed information is listed in Table 1.

	Google Map	Qpides!	Indoor Server
Information for storing	GPS Coordinates	Coordinates of the restaurant	Pointing positioning on the restaurant's map
	Guide to the destination	Data information of signals WIFI, Bluetooth, Cell-mobile	
		Map of the restaurant	
Information for processing	Destination Restaurant's coordinates from Qpides	Destination user's coordinates from Google Maps	Map of the restaurant with the map's mapping signals from Qpides
	Destination Restaurant's coordinates chosen		Signals of WIFI, Bluetooth, Cell-mobile from user terminal

Table 1. Use of information by servers

The three servers work at the same time and they will depend on the instructions of the user or the requirements of the restaurant.

It is possible to identify which server is acting analyzing the behavior of Qpides!'s users.

When the user starts the app, Qpides! redirects him to the Google Map-server where the user sees a map with his position. Then, if the user wants to know the nearby restaurants, the user sends his request to the Qpides!-server using his actual position. Qpides! server returns the GPS coordinates, taking into account the radius of searching selected. With these GPS coordinates, different restaurants appear in the Google map. When the client selects a restaurant, Google Maps helps to identify it. At the same time Qpides!-server stores this selection and send to the user-app the SSID of the establishment. The SSID helps the app to identify the restaurant when the client is there.

When the user arrives at the restaurant the app compares the SSID sent by the Qpides! Server with the establishment's SSID and if they match the app switches from the Google-server to the Qpides!-server and Indoor server. The position will be determined by the Indoor-server but the coordinates of the table and the menu is determined by the Qpides!-server. At this point the Google-server will no longer be involved until the client restarts searching.

The positioning inside the restaurant, along with the order and the number of the table will be used to make the ticket, and this information will be stored on the Qpides!-server.

When the client pays the bill, the app erases the SSID of the restaurant and all information in the user device.

The Table 2 shows the interaction between the user's instruction and the servers.

USER-APP	Google Map server	Qpides! server	Indoor server
1. Starts the Qpide! app	X		
2.Views the possible restaurants	X	X	
3.Selects the restaurant	X		
4.Goes to the restaurant	X	X	
5.Arrives to restaurant		X	X
6.Goes to the table			X
7.Make the order		X	X
8.Pay the bill		X	X

Table 2. Server and user instruction

A similar situation happens with the restaurants. When a client books a table in a restaurant, the restaurant receives the location of the client. This information is stored in the Qpides!-server. Once this information is stored, the restaurant may calculate the arrival time of the client or monitoring the client route. When the user arrives to the restaurant the Indoor-Server assigns the table. The Indoor-server provides the restaurant the management of the table's distribution. When the customer fails to arrive at the restaurant, the server QPides! free booking and records the incident.

The table 3 shows the interaction between the restaurants needs and the servers.

RESTAURANT-QPIDES	Google Map server	Qpides server	Indoor server
1.Received a booking		X	
2.Monitoring the route of the client (possible)	X	X	
3.Estimating the arrival time to the restaurant (possible)	X	X	
4.Fail booking		X	
5. Assigning the client's table			X
6.Receiving the order		X	
7.Colleting the ticket		X	X

Table 3. Server and the Restaurant needs

1.3. The entrepreneur

QPides! is born from the association of 3 entrepreneurs who are willing to implement and achieve the goals of the company's motto. Such entrepreneurs' data is provided:

- **Emilio J. Perez Salgado**
  - Tec. Telecommunications Engineer
  - 8 years in Radio Communication
- **Ivan BalboteoToledano**
  - Tec. Telecommunications Engineer
  - 2 years in Smart Systems projects
- **Juan Carlos Arco Fernandez**
  - Tec. Telecommunications Engineer, Msc Digital Electronics, Msc TIC Management
  - 7 years in space projects R+D

The highly telecommunication based background of the entrepreneurs, makes the project reliable as there would be no issues in such scheme.

The additional educational and experience in similar environment (business/project based management) increments even more the successful development of the business proposal. A demonstration of such background is the environmental analysis made to evaluate the impact of such variables in the project.

1.4. The environmental

The success of any application development is highly dependent on the state and trend of different factors, such as:

- **Social:** There is an increasing bargain power on clients as the social intelligence is used. Such power can also establish new trends when looking for restaurants and other leisure solutions.
- **Economic:** The current crisis demands for solutions with low investment and maintenance while providing the best quality possible. It is needed a higher control on expenses to maximize the competitiveness between restaurants.
- **Technologic:** The diffusion and extended use of smart devices (especially smartphones) is a fact. The versatility of these devices makes it possible to access/manage information in real-time. Such information would determine the selection of the restaurant where to have lunch.
- **Institutional:** There are still some legal requirements to be implemented, accordingly to the new technological solutions. These requirements, though, are expected to be defining in a near future.

CHAPTER 2. Indoor location solution

Following definition [10], a Technique is "*the way of solving or doing a thing*". However, technology is "*the application of practical sciences to industry or commerce. It is the methods, theory, and practices governing such application: a highly developed technology. It is the total knowledge and skills available to any human society for industry, art, science, etc.*"

Currently, there are several tools on the market for indoor location. Also, there are different technologies to determine the indoor positioning like Wi-Fi triangulation, Wi-Fi Fingerprint, Beacons, Bluetooth, Sensors, Indoor Lights, Magnetic Fields, etc. and different technics too, like Google-indoors. There are trade-offs to each of them.

Wi-Fi is low cost and easy to find in any place, but unfortunately not very accurate. High precision location usually requires higher cost and infrastructure. Proprietary technologies can be very accurate, but more expensive and not as usual; so apps base on that technology only work where this expensive infrastructure is installed. High accuracy today is considered to be 1 to 5 meters. Medium accuracy is 6 to 10 meters, and low accuracy is over 11 meters.

2.1. Technology review

In this section, I analyze the different technologies to determine the indoor positioning. [18]

Wi-Fi Triangulation – Wi-Fi Triangulation [4] measures loss or strength signals from three or more Wi-Fi APs to triangulate position. The app doesn't need to access the Wi-Fi; it just pings to measure signal strength. Some systems that use this technic are Ekahau, Meridian, Navizon, Proximus Mobility, Sense Where and WifiSlan.

Wi-Fi Fingerprinting - Smartphones turn on Wi-Fi for a few seconds to get a Wi-Fi Fingerprint [4] and associate it with a Check-In location. Compares the current Wi-Fi Fingerprint to a known database of Fingerprint/Location pairs. Some systems that use this technic are Aisle 411, EveryFit, Guardly, Indoo.rs, Insiteo, PoleStar, PointInside, Qubulus, RedPin, WalkBase, Wifarer and Yfind.

Beacons - Cheap, low power, radio beacons located at known positions within a building. Could be Bluetooth, High frequency radio, radio inference or other proprietary radio signals. Uses the same location triangulation methods as Wi-Fi. Some systems that use this technic are BlinkSight, Ekahau, Insiteo, InvisiTrack, Locata, OmniSense, Quuppa, RedPin, Teldio, UbiSense WiseSec and Zulu Time.

Bluetooth - Many electronic devices contain Bluetooth radios, including every smartphone. Bluetooth sensors can read signals from dedicated beacons, or dynamically create a mesh network of Bluetooth signals that

refines location. Some systems that use this technic are Newaer, PoleStar, Proximus Mobility, RedPin, Proximus Mobility, Quuppa and Sense Where.
Sensors (Accelerometer, Gyro, Compass, Barometer, etc.) - Most smartphones contain multiple sensors that can measure your direction, turns, speed, and height above sea level to create a three dimensional view of your location. Starting with a known position from other methods such as GPS, cellular, or Wi-Fi, the smartphone sensors can be used to track your position inside a building. Some systems that use this technic are Aisle 411, BlinkSight, EveryFit, IndoorGo, Loctonix, Movea, NaviSens, PointInside, SenseWhere and TRX Systems.

Indoor Lights - LED lights in the ceiling can be programmed to pulse in milliseconds, so fast the human eye can't detect the pulse. Your smartphone camera can detect the pulses and distinguish between different lights and triangulate your position. A system that uses this technic is ByteLight.

Magnetic Field - Magnetic sensors can pick up the Earth's natural magnetic forces to determine latitude /longitude position similar to the way a compass works, but two dimensional, and much more accurate. Some systems that use this technic are Indoor Atlas and Indoors.

Cell Tower Signal – Triangulates approximate position using cell tower signal strength. Some systems that use this technic are Artilium, Combian, Insiteo and RedPin.

Low Orbit Satellites – Works like GPS, but lower orbit, higher power signal can penetrate inside buildings. A system that uses this technic is NextNav.

Camera Technology - A ceiling or wall-mounted camera within a building compares auto-generated snapshot photos from your Smartphone. Object recognition software uses pattern matching to compare those smartphone snapshots to the wall-mounted camera to determine precise location. Some systems that use this technic are WhereLab and Omiimii.

Thinking on the restaurant application the Table 4 presents the weakness and the strengths of the technologies reviewed on point 2.2.

Technologies	Weaknesses	Strengths
Wi-Fi Triangulation	High cost. More AP. Wi-Fi behavior	Easy implementation
Wi-Fi Fingerprinting	Wi-Fi behavior	Low cost, more flexible
Beacons	High cost. External devices	Good accuracy
Bluetooth	High cost. External devices	Good accuracy
Sensors	High cost. External devices	Good accuracy
Indoor Lights	External devices	Good accuracy
Magnetic Field	High cost. External devices	Good accuracy
Cell Tower Signal	Bad accuracy	Low cost
Low Orbit Satellites	Difficulties to implemented	Good accuracy
Camera Technology	High cost. External devices	Good accuracy

Table 4. Technologies weaknesses and strengths

There are two options for make the indoor-localization using Wi-Fi signals, the triangulation and the fingerprinting. Triangulation involves mapping signal strength as a function of distance while fingerprinting creates a probability distribution of signal strengths at a given location [3].

Wi-Fi triangulation's goal is to map power (RSSI) as a function of distance. This method requires a steep linear characterization curve in order to be properly implemented. Functions describing these curves are then used with live power values as input to generate coordinates location prediction (X,Y).

Wi-Fi Fingerprinting creates a map of an area based on the power data from multiple access points (APs) and generates a probability distribution of power values for a given coordinates location (X,Y). Live power values are then compared to the fingerprint to find the closest match and generate a predicted (X,Y) location.

We decided to work with fingerprint instead of triangulation because triangulation needs the control of the AP [4] for determine the distance as a function of the power transmission. Without the knowledge of power transmission and the position of AP, it was not possible to determine the distance. So, if we used our own APs, it would be necessary to buy it, and from a commercial point of view it won't be desirable for the Qpides! app. Also, the fingerprint only needs the power mapping area (sniffing the Wi-Fi signals).

2.2. Considered Indoor location Wi-Fi methods

The cheapest and simple implementation for this project was working with Wi-Fi signals or using an existing technic. Following this, to do the positioning inside a restaurant, the project revealed two different ways.

The first one is the Google-indoor app and the second one analyzes RedPin and Insiteo tools, based on fingerprint technology.

2.2.1. Google indoor

Google [5] has the tool Google-maps. The user can upload the map of the building, and train it. Google can store the information regarding to the signals in a specific point of the building's map. The developer/user, using a device with Google maps and activating signals access that wants to send (Wi-Fi, 3G,LTE, GPS, Bluetooth), goes around the surface that would like to map and send for each specific position point. With this information, the user sends the signal information (name of SSID of the Access Nodes, GPS points, the Phone Cell).

Google makes a route over the map. This route is doing it by the information of the signals and the knowledge of devices than exist inside the floor. Google provides a tool-developer[1], for uploading the indoor map that they want to use in Google server map.

[1] http://www.google.com/intl/es-419/maps/about/partners/indoormaps/

The key element of Google in front of other solutions is that it provides positioning inside and outside of a building. The weakness of this technic is that it's necessary to pay a quote if a user uses the Google's tool in their App. It has a free usage for developers of 28 day for application.

We focused on Google-indoor thinking that it would be the best solution for the restaurant application. Therefore, once you are registered as developers, you can follow the steps that we outlined in the Google page with tutorials.

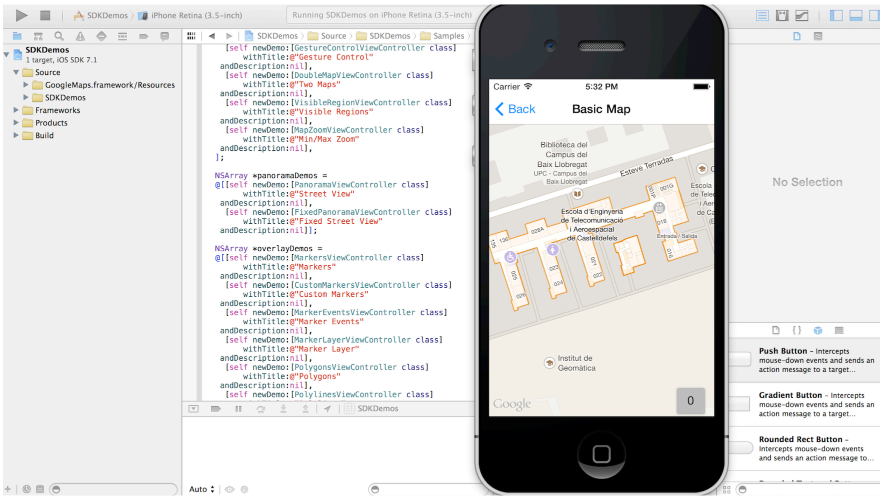


Figure 3. EETAC map in Xcode

We work on an App for mobile, in particular for IPhone using Xcode, the developer tool that provides apple for IOS. See Figure 3. The purpose was to test the Google Tool with a device.

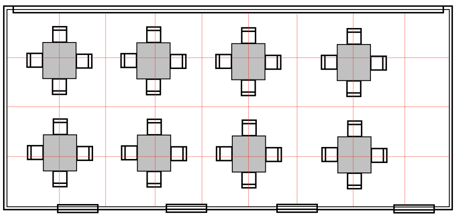


Figure 4. Restaurant Map for Lab 016

We focused on the uploading map in Google server to start our test. The main idea was obtain a map like shows on Figure 4 incrusted on the lab 016 for train it and analyze the results. But none of the 9 maps got on the Google sever were visible on Google maps and therefore could not test the application for interiors.

After sent a mail to Google explaining the problems and no having answered we decide discard Google solution.

2.2.2. **Insiteo**

Other solution that we analyze was Insiteo. Insiteo provides a free App for developers to use it on user part, but Insiteo Company is the owner of the server. The developer puts in the server the map of the surface that he wants control and sends de AP information.

Insiteo as RedPin, coverage the access-point areas than has implemented on the server. Insiteo demands the payment of the usage of the server[2]. It has a trial-time of 30 days.

Insiteo was discarded because it worked like RedPin; but with the handicap that the server had to be paid.

2.2.3. **RedPin**

RedPin [1] has two basic components: a sniffer component that gathers and collects information about different wireless devices in range in order to create a fingerprint (App), and a locator component, which stores measured fingerprints in a repository and contains the algorithm to locate a mobile device (server).

RedPin can determine the location of a device with room level accuracy. Being a fingerprint-based system, RedPin will not provide geographic coordinates but rather symbolic identifiers as for example the number or name of a room.

The interval scanner runs in the background, and periodically performs a scan of Wi-Fi networks, creating fingerprints by attaching the current location to the new Wi-Fi measurements. The fingerprint is then being sent to the server and is stored there. Interval scanning is stopped whenever a significant movement is detected through the built-in phone accelerometer, or when the users try to either add a new location or locate themselves.

RedPin only provides positioning in the surface than has an access-point mapping. The weakness of RedPin is that provides an accuracy of a room and for our final application we need more accuracy. In other way, the development of RedPin stopped on 2010 and is designed for worked only with access-point signals. The goodness is that RedPin is a free tool.

At this point we put our effort on the RedPin application based on fingerprint technology.

The first idea was changing the way of operating. We focused on how RedPin works to understand the fingerprint technology instead of working in App, like we did when we work on Google solution. The Appendix E shows an overview of RedPin AP.

We assume that RedPin, like exposes on the web site [1], has a room-accuracy. This was the reason that we discarded use for the restaurant application this tool. But we based on RedPin functionality to try to analyze our fingerprint system.

---

[2] This is the reason that only works with Google and RedPin solution.

CHAPTER 3. Experimental Scenario Setup

Although the study is oriented to analyze indoor location within a restaurant, it was not possible to have access to a real restaurant dinning room. So, we decided to use an alternative location. For that, we decided to use a room within EETAC's D4 first floor building as experimental scenario.

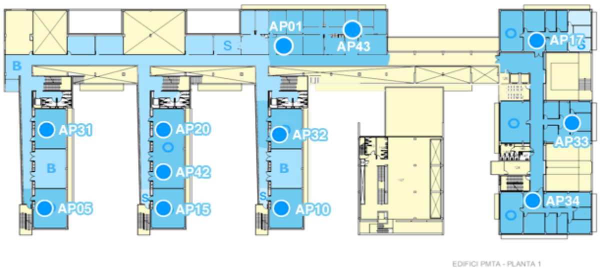


Figure 5. EETAC AP floor 1

The Figure 5 shows de distribution of APs in floor 1. Whole floor, we chose the room 130 that has 72m^2. The Figure 6 shows the room 130 where we make the fingerprint mapping.

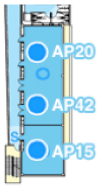


Figure 6. Room 130-133

This room does not have a real restaurant layout, so a virtual distribution is chosen as the one presented in Figure 7. We assume four people for table. It's an easy distribution to explain the mapping concept and the fingerprint idea.

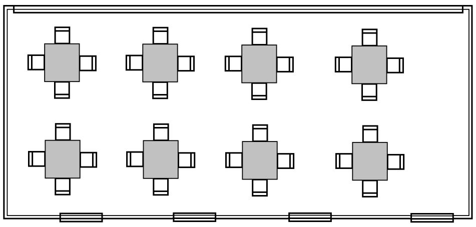


Figure 7. Table distribution in room 130

To perform the mapping of the area, we need to know what kind of signals we are going to work. Even though RedPin can work with different signals we only, do the mapping with Wi-Fi signals.

It is necessary take in to account that the use the combination of more different signals, results a better accuracy. That means that the combination of GSM with Bluetooth and with Wi-Fi increases the accuracy in front of Wi-Fi signal alone.

3.1. Measurement instrumentation

To collect the Wi-Fi data in room 130 we use three devices and specific software for each of them.

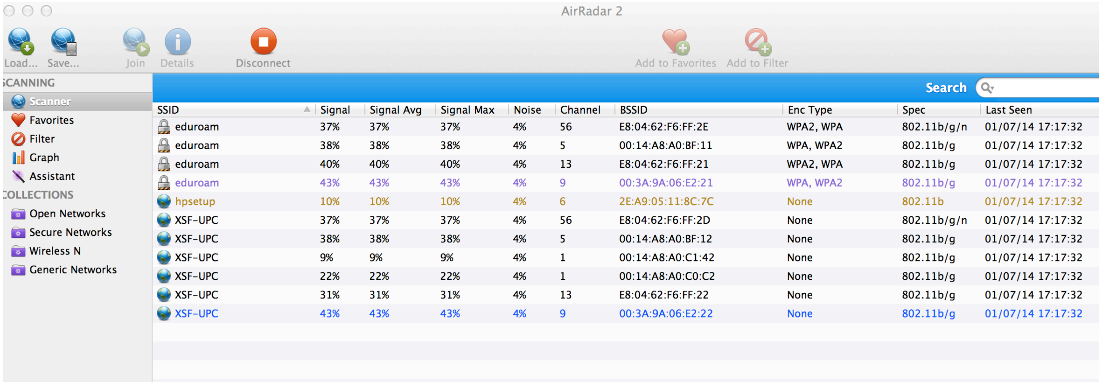


Figure 8. AirRadar

We use software called AirRadar [6] for MacBookPro, software WIFI SCANER for Iphone5S and inSSIDer for Acer-PC with Windows XPSP1.

The software allows us to get the Wi-Fi signals received at a given point. Thus, provides information on each AP in terms of power, SSID, BSSID, noise, type of encryption and the transmission channel. The Figure 8 shows AirRadar's example used to obtain the data power by MacBook.

We estimate for our scenario, that the optimal number of users could be 16 with different mobile phones, acquiring a 2 hours of data each one. This estimation takes into account the most relevant mobile phones on the market [3]. But we don´t have the possibility of bring 16 different devices to make the test. This estimation was doing taking into account the mobile's manufactures.

3.2. Measurement procedure

The next point is to make an imaginary distribution on the room like it shows on Figure 9. The different areas will be the points were we make the Wi-Fi measurements and we store the information in database. All of the results are included inside the Annex A. These results help us to determine the fingerprint on the restaurant (room 130).

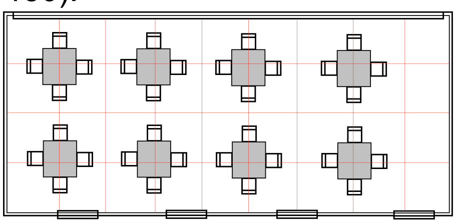


Figure 9. Room measurement areas

At the end, we have coordinates areas (X,Y) with the information of the AP for each area. The separation between X coordinates is 1,5 meters and the Y coordinates is 1 meters. Like it shows on Figure 10, we achieve the mapping of the room.

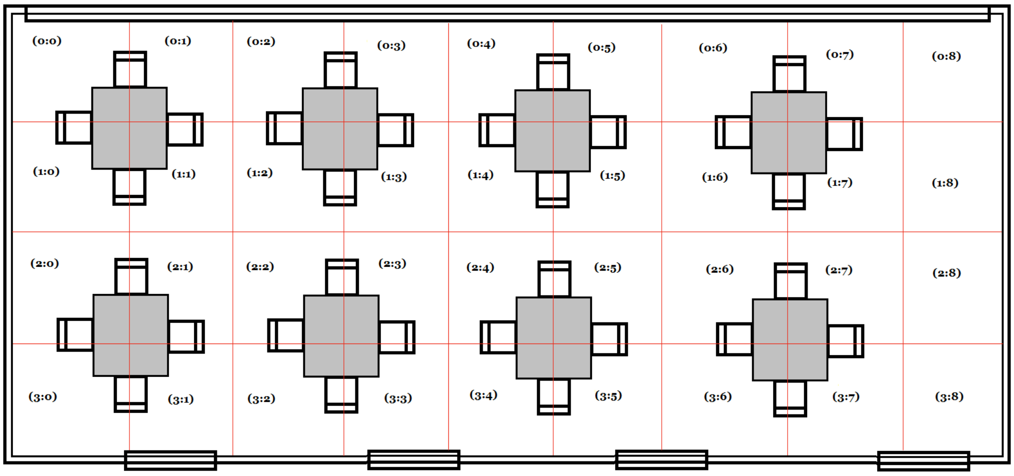


Figure 10. Mapping room 130

The **Table 5** shows the lecture of the (0,0) area. It is important to highlight that the SSID is the same for the entire App, for this reason we work with the BSSID to distinguish between APs.

(0;0)	SSID	Signal	Signal Avg	Signal Max	Noise	Channel	BSSID
	XSF-UPC	11%	11%	11%	4%	1	00:0F:24:D6:A2:E2
	XSF-UPC	36%	36%	36%	4%	5	00:14:A8:A0:C0:A2
	XSF-UPC	10%	10%	10%	4%	13	08:D0:9F:17:30:02
	XSF-UPC	11%	11%	11%	4%	9	08:D0:9F:86:64:F2

Table 5. Mapping information (0;0) area

Using the method described, the whole experimental scenario has been characterized using a total of 12600 measures from three devices in 36 different areas. These measures only considered the power levels of 3 APs (the rest were discarded). The use of these measures is discussed in Chapter 4.

3.3. Room considerations

To analyze the power distribution on the room 130 (the restaurant) we take in consideration 3 AP. For each of them will make the mapping area in terms of power. The three BSSID are: E8:04:62:F6:C4:82, 2C:3F:38:C1:BC:FD and 00:14:A8:A0:C0:A2.

The selection of APs gave coverage around the room. Every zone has distributed around 14,5 m and are delimitated by 1 meter. At the end we will have a matrix map like is showed on Figure 10.

3.4. Room Power behavior

To create the fingerprint we analyze the behavior of each AP around the entire map independently for the different devices (Windows-PC, Smart phone and MacBook). We consider de power distribution for the four zones taking into account the median power value. In this section we present MacBook's results. The remaining results are on Appendix A.

Now, with this information we elaborate a simple fingerprint. This means that we only consider the mean-power distribution of one AP. The idea is analyzed if is possible with one AP localize a user in the restaurant.

We start with the AP 00:14:A8:A0:C0:A2. Like we see on Figure 11 the power has a good spread around the room. But if we use only this access-point we will have some problems for positioning the user at 1,5 meters were the power of zone 2 and zone 3 are the same. The same problem appears in 3 meters, 6meters, 9 meters, 10meters and 12 meters. But it is important focused that in these critical points only be overlapping two zones for each point. This will be important to elaborate the final fingerprint. Other problem is that in same zone we have the same power, this is visible for all the zones.

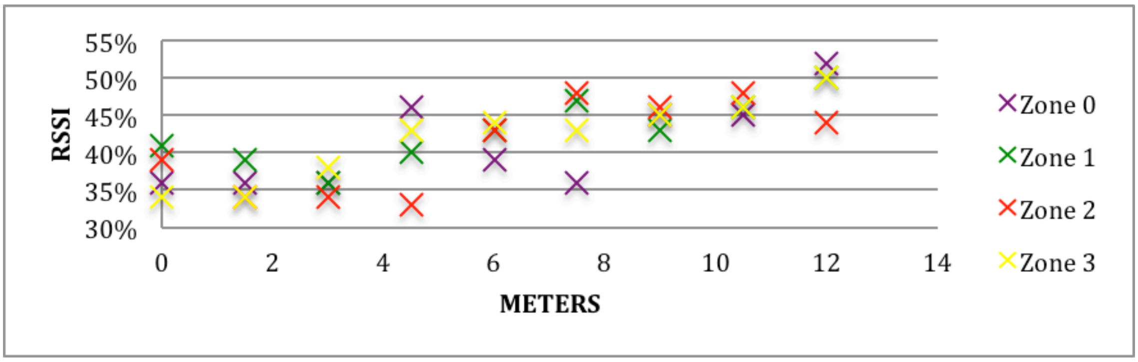


Figure 11. RSSI distribution map of 00:14:A8:A0:C0:A2

We going to analyze the AP 2C:3F:38:C1:BC:FD like the first one, we consider the power distribution for all the room. We expect a similar behavior to the previous case. As we see in Figure 12, in this case overlap along the entire map. Although this did not discard the AP because the power levels was higher than in the previous AP. We can use this skill to develop the fingerprint.

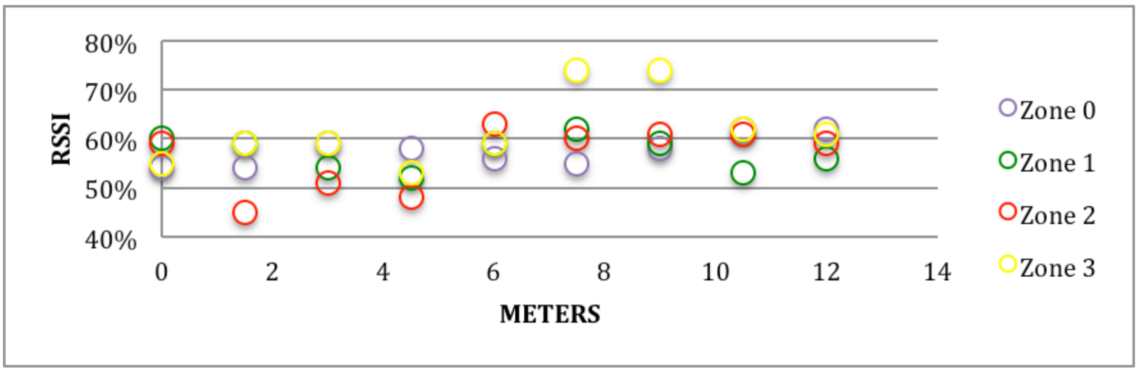


Figure 12. RSSI distribution map of 2C:3F:38:C1:BC:FD

Finally, like we do in the previous cases, we analyze the last AP E8:04:62:F6:C4:82.

As we see on Figure 13 presents the best power distribution. It is possible distinguish the different zones around the map. But it is not possible use this AP alone because exist some points that are very nearly like we see at 7,5 meters or 10,5 meter.

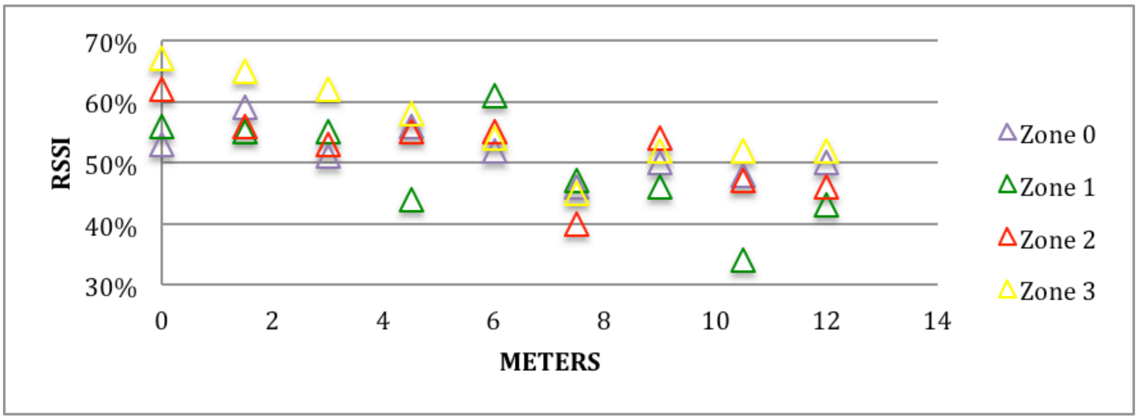


Figure 13. RSSI distribution map of E8:04:62:F6:C4:82

3.5. APs combination

The fingerprint technique was used to generate a coverage pattern obtained from the combination of two APs received power data. This pattern is later used with clients' data to determine their location inside the room.

Our scenario contains three different APs, so the fingerprint technique is applied for each of the APs possible combination. Each obtained power pattern distribution in the study zone is considered in order to choose the most adequate pair of APs.

The next pictures show the three possible combinations and the mean-power distribution around the restaurant. This point shows only the MacBook case. The combination using a Windows-PC and IPhone 5S appears on Appendix C

The combination shown on the Figure 16 and Figure 14 presents more overlapping than Figure 15. A same situation happens on IPhone 5S case. But the Windows-PC combination did not presents a clear distribution instead presents high overlapping in all combination.

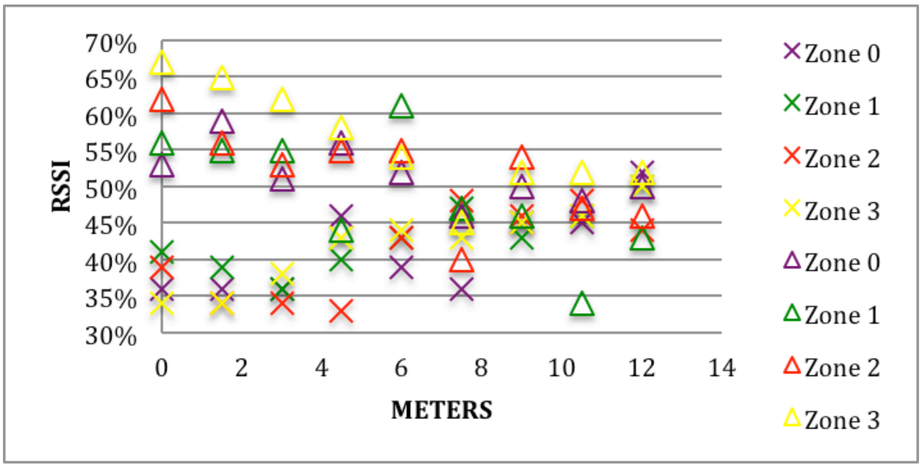


Figure 14. Fingerprint 00:14:A8:A0:C0:A2 and E8:04:62:F6:C4:82.

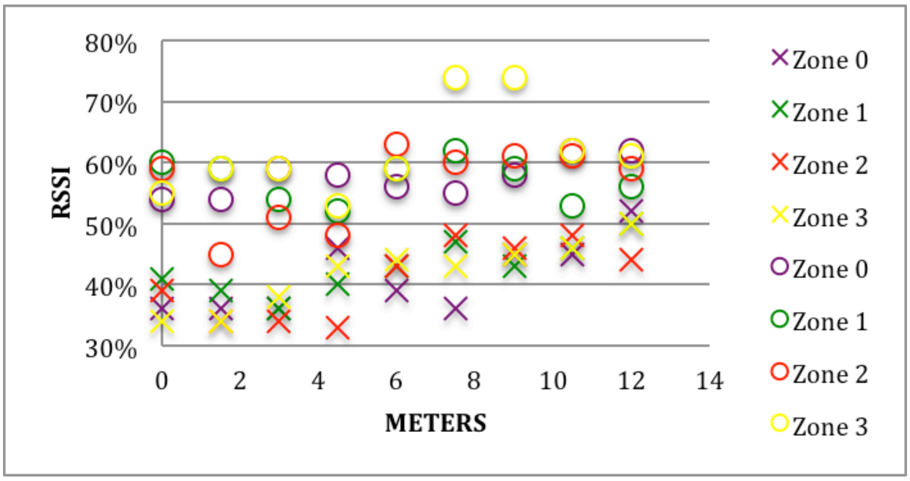


Figure 15. Fingerprint 00:14:A8:A0:C0:A2 and 2C:3F:38:C1:BC:FD.

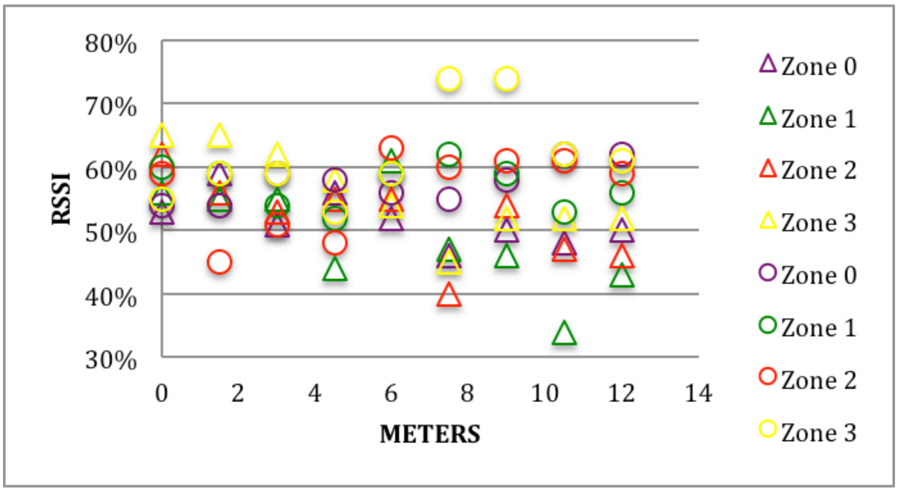


Figure 16. Fingerprint E8:04:62:F6:C4:82 and 2C:3F:38:C1:BC:FD.

The analysis to choose the AP was made using the average of the data measured on the room 130 for the three devices.

According with the results, the AP selected to elaborate the fingerprint are

00:14:A8:A0:C0:A2 and 2C:3F:38:C1:BC:FD.

3.6. **Finger print results.**

Once selected the AP to develop the fingerprint, we focus on analyzing the behavior taking into account all data measured. For this we use a box plot where quartiles are used 12.5%, 25%, 50%, 75%, 87.5% of all measured data.

This type of diagram can clearly shows the behavior of the data obtained in an area. At the same time, it shows the interaction with the other zones.

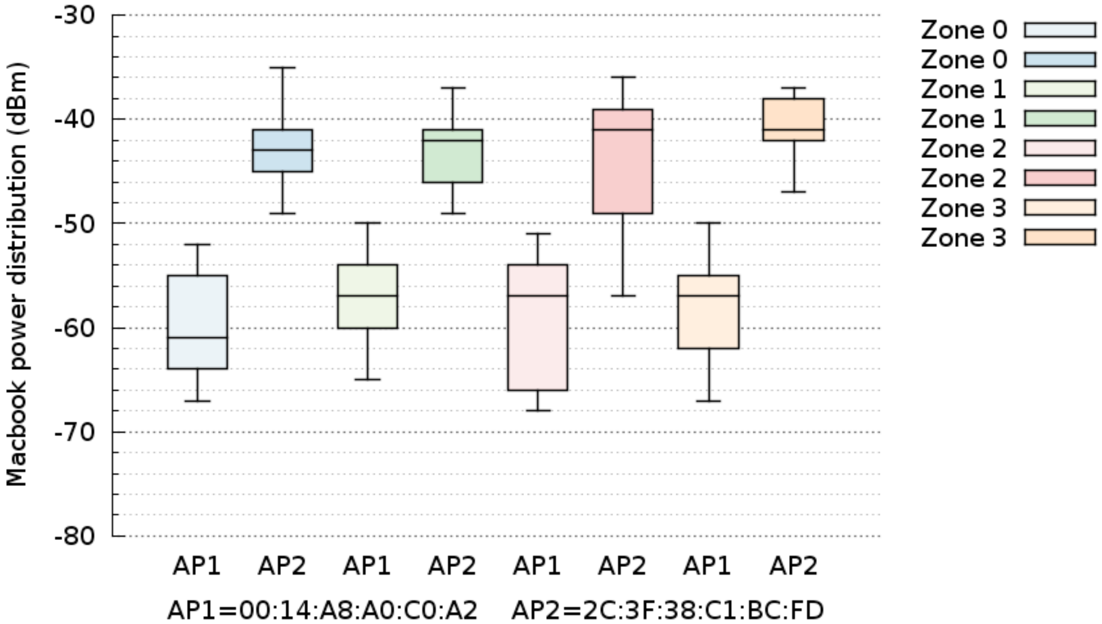


Figure 17. Box plot data for MacBook on room 130

Figure 17 shows the MacBook's box diagram obtained from 1800 data/AP. These data are presented in terms of the power per zone and per AP. It is possible note that between AP is a good separation and low dispersion in data.

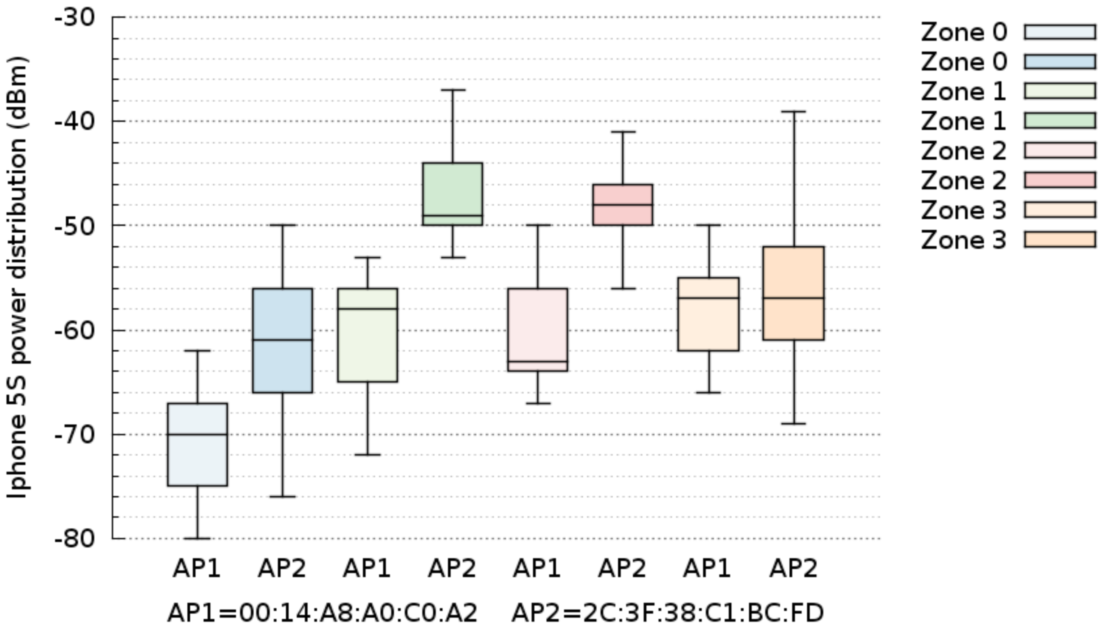


Figure 18. Box plot data for IPhone 5S on room 130

Figure 18 shows the IPhone 5S's box diagram obtained from 1800 data/AP. These data are presented in terms of the power per zone and per AP. It is possible note that between AP is a good separation and high dispersion in data. In zone 0 the AP2 presents a data power between -50 dBm and -76 dBm with a margin of 26 dB. This is significant in comparison with 14 dB presented by the MacBook case for the same AP in the same Zone.

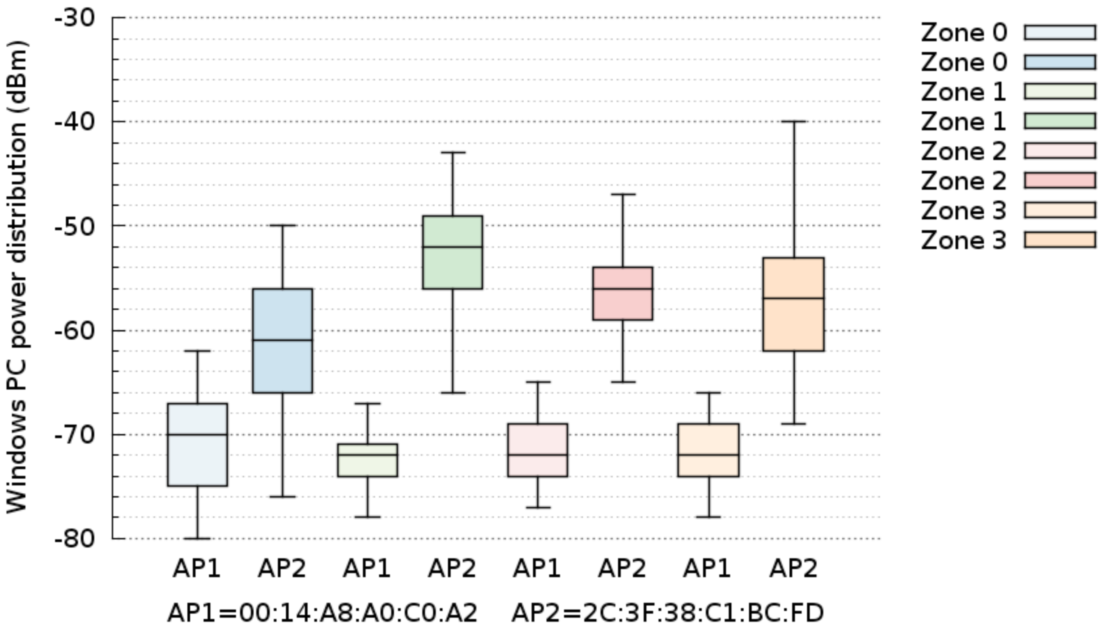


Figure 19. Box plot data for Windows-PC on room 130

Figure 19 shows the Windows-PC's box diagram obtained from 1800 data/AP. These data are presented in terms of the power per zone and per AP. It is possible note that between AP is a good separation and high dispersion in data.

We can see that the combination of the two measures is not unique to an area. Is possible to obtain the same reading of AP1 and AP2 for zone 1, zone 2 and zone 3. This fact happens on all three devices, as shown by box plots.

CHAPTER 4. Evaluation

The fingerprinting uses the map and the data distributions to predict a location. We use Weka to evaluate and simulate the data obtained.

Weka [10] is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well suited for developing new machine learning schemes.

4.1. Prediction Methods

Two prediction methods used to determine the location of the node based on the fingerprint data are Nearest Neighbor and Markov Localization.

4.1.1. Nearest Neighbor

The Nearest Neighbor (NN) [8] method simply calculates the Euclidean distances between the live RSSI reading and each reference point fingerprint. The minimum Euclidean distance is the Nearest Neighbor and the likely (X,Y) location.

The Euclidean Distance is represented by $\sqrt{\sum_{i=1}^{6}(R_i - FP_i)^2}$ (5.1)

Two versions of Nearest Neighbor are used: unconstrained search-space and constrained search-space. Unconstrained search-space looks at the entire fingerprint map to find the closest match. Constrained search-space only searches within a given distance from a previously predicted location.

The idea is that a moving object can only travel up to a maximum distance from its previous location within the time it takes to collect a live RSSI reading and searching through the entire map is unnecessary. This also has the effect of ignoring predicted locations that are closer based on Euclidean distance but physically impossible as the next location based on the previous location.

The RedPin algorithm is a deterministic algorithm call k-Nearest Neighbor (kNN) [2] with k=1. To predict the location of a target Wi-Fi measurement it calculates an ”AP-coincident” level between the target measurement and every measurement stored in the database.

The location of the most similar measurement from the database is used as the output for the algorithm. Every AP seen in both the target and a database measurement contributes a bonus factor to the similarity level. Every AP not shared by both measurements receives a penalty value that is reduced from the similarity level. In addition, for every matching pair of APs a signal contribution is calculated based on similarity between their power values (RSSI) and added to the total similarity level.

4.1.2. Markov Localization

Markov Localization makes use of the statistical data of the fingerprint to guess the most likely position [4]. This is done in two steps: Prediction and Correction.
Prediction Step:

$$\rho(L_t) = \sum_{L_{t-1}} \rho(L_t|L_{t-1})\rho(L_{t-1}) \quad (5.2)$$

$\rho(L_t)$ is the probability of being at location L at time t.
$\rho(L_t|L_{t-1})$ is the probability of being at location L at time t given the previous location L at time t-1.

This effect restricts the search-space to the most likely region based on what is physically possible based on what we know about the object's motion.

Correction Step:

$$\rho(L_t|R[]) = \rho(R[]|L_t)\rho(L_t) * N \quad (5.3)$$

$\rho(L_t|R[])$ is the probability of being at location L at time t given the RSSI values R[] we received at time t.

$\rho(R[]|L_t)$ is the probability of having RSSI values R[] given a location (the probability density function generated from the fingerprint data) and $\rho(L_t)$ is the probability of being at that location (from prediction step).
N is a normalization factor.

4.2. Nearest Neighbor k=1

Weka allows us to train and test the data using the kNN algorithm. At the same time provides different ways to display and process the results.

The first analysis of a test consisted in training and testing with a same device (i.e. Weka). This means that, using Weka software, we used the 1800 data for each device to train the area map. Then, we used the same data for simulate that a user was inside a restaurant so we were able to analyze his behavior. The evaluation was done it using kNN algorithm with k=1. The result shows on Table 6 for the three devices.

Evaluation on training set	**MacBook**		**IPhone 5S**		**Windows-PC**	
Correctly Classified Instances	1411	78.389%	1478	82.111%	1396	77.556%
Incorrectly Classified Instances	389	21.611%	322	17.889%	404	22.444%
Kappa statistic	0.7777		0.816		0.7691	
Mean absolute error	0.0151		0.0129		0.0162	
Root mean squared error	0.0866		0.08		0.0895	
Relative absolute error		27.915%		23.886%		29.970%
Root relative squared error		52.680%		48.653%		54.446%
Total Number of Instances	1800		1800		1800	

Table 6. Summary evaluation on training set

Taking a look in the before summary table, it can determined that for 1800 instance for each device in MacBook it had a 1411 correctly instance, in Iphone-5S 1478 correctly instance and in Windows-PC 1396 instance.

Other interesting data reference in table summary is the Kappa statistic [9].
The equation for κ (Kappa) is:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{6.1}$$

Where Pr(a) is the relative observed agreement among raters, and Pr(e) is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly saying each category. If the raters are in complete agreement then $\kappa = 1$. If there is no agreement among the raters other than what would be expected by chance (as defined by Pr(e)), $\kappa = 0$.

The κ for the MacBook was 0.7777, for IPhone was 0.816 and for Windows-PC was 0.7691. That means that the algorithm had a good accuracy when determine the position of user in the training area. In concrete for all devices is around to 80% of success. It was possible say that this test worked satisfactory with a high success.

The next test consists in training with one device and testing with other one. The Table 7 shows de results obtained by the different training devices.

Training device	**MACBOOK**				**IPHONE 5S**				**WINDOWS-PC**			
Testing device	**IPhone 5S**		**Windows-PC**		**MacBook**		**Windows-PC**		**IPhone 5S**		**MacBook**	
Correctly Instances	91	5.06%	45	2.50%	99	5.50%	103	5.72%	52	2.89%	66	3.67%
Incorrectly Instances	1709	94.94%	1755	97.50%	1701	94.50%	1697	94.28%	1748	97.11%	1734	96.33%
Kappa statistic	0.0234		-0.0029		0.028		0.0303		0.0011		0.0091	
Mean Absolute error	0.0528		0.0542		0.0527		0.0524		0.054		0.0535	
Root mean Squared error	0.2246		0.2296		0.2221		0.2268		0.2298		0.2302	
Relative Absolute error		98%		100.30%		97.53%		97%		99.90%		99.09%
Relative Squared error		136.66%		139.69%		135.18%		138%		139.84%		140.07%
Total Instances	1800		1800		1800		1800		1800		1800	

Table 7. Summary of training and testing

We were considering the same point than the earlier test, where the testing and training was do it by the same device, we can assume that the usage of a device inside a map training with other type of device didn't works.

The values of κ (Kappa) were equal or lower than 0. Therefore, we conclude that the predictive algorithm does not work as desired. It proves that it is not suitable for this application. According with the results the prediction process would be more successful guessing a coin toss.

4.3. kNN algorithm for different k value

Taking into consideration the worst case (training Windows-PC and testing with MacBook) we will analyze the behavior changing the k value of the algorithm kNN.

	k=3		k=5		k=6		k=7		k=10	
Correctly Instances	90	5%	126	7%	136	8%	140	8%	120	7%
Incorrectly Instances	1710	95%	1674	93%	1664	92%	1660	92%	1680	93%
Kappa statistic	0.0229		0.0434		0.0491		0.0514		0.04	
Absolute error	0.053		0.0523		0.0522		0.0523		0.0524	
Mean squared error	0.2248		0.2222		0.2215		0.2215		0.2213	
Relative absolute error		98%		97%		97%		97%		97%
Root relative squared error		137%		135%		135%		135%		135%
Total Instances	1800		1800		1800		1800		1800	

Table 8. Results changing the K value worst case

As shown in Table 8, increasing the value of k doesn't be a significant improvement.

Taking into consideration the best case (training with IPhone 5S and testing with IPhone 5S) we will analyze the behavior changing the k value of the algorithm kNN

	k=3		k=5		k=6		k=7		k=10	
Correctly Instances	1450	81%	1423	79%	1415	79%	1404	78%	1364	76%
Incorrectly Instances	350	19%	377	21%	385	21%	396	22%	436	24%
Kappa statistic	0.8		0.7846		0.78		0.7737		0.7509	
Absolute error	0.0138		0.0146		0.015		0.0152		0.016	
Mean squared error	0.0832		0.0863		0.087		0.0883		0.0916	
Relative absolute error		25%		27%		28%		28%		30%
Root relative squared error		51%		52%		53%		54%		56%
Total Instances	1800		1800		1800		1800		1800	

Table 9. Results changing the k value best case

As shown in

Table 9, when increasing the value of k the test result is worse.
Now we test the behavior using a different algorithm to compare with kNN.

4.4. Supervised learning methods

Supervised learning is a technique for deriving a function from training data. Training data consists of pairs of objects, usually vectors. One component of the pair is the input data and the other desired results. The output of the function can be a numeric value or class label.

The objective of supervised learning is to create a function able to predict the value for any valid input object after having seen a number of examples, the training data. For this you need to generalize from the data presented to previously unseen situations. In this way we analyze SVM algorithm and Bayesian algorithm, using our data device obtained by the AP-1 and AP-2 and compared it with the results obtained using kNN algorithm, shows on Table 6.

4.4.1. Support Vector Machines algorithm

An algorithm based on Support Vector Machines (SVM) builds a model to predict whether a new point (whose status is unknown) belongs to one category or the other.

The Table 10 shows the result using a SVM, in concrete a SMO (Sequential Minimal Optimization) algorithm provides by Weka. [10]

Evaluation on training set	MacBook		IPhone 5S		WindowPC	
Correctly Classified Instances	1170	65%	1232	68%	1030	57%
Incorrectly Classified Instances	630	35%	568	32%	770	43%
Kappa statistic	0.64		0.6754		0.56	
Mean absolute error	0.0525		0.0525		0.0526	
Root mean squared error	0.1606		0.1606		0.1607	
Relative absolute error		97%		97%		97%
Root relative squared error		98%		98%		98%
Total Number of Instances	1800		1800		1800	

Table 10. Test result using a SVM algorithm

4.4.2. Bayesian algorithm

The Table 11 shows the result using a Bayesian algorithm, in concrete a Bayes-Network algorithm provides by Weka. [10]

Evaluation on training set	MacBook		IPhone 5S		WindowPC	
Correctly Classified Instances	1399	78%	1410	78%	1322	73%
Incorrectly Classified Instances	401	22%	390	22%	478	27%
Kappa statistic	0.7709		0.7771		0.7269	
Mean absolute error	0.0206		0.0196		0.0256	
Root mean squared error	0.0929		0.0904		0.1037	
Relative absolute error		38%		36%		47%
Root relative squared error		56%		55%		63%
Total Number of Instances	1800		1800		1800	

Table 11. Test result using a Bayesian algorithm

If compare the results with the kNN of Table 6 we see that the behavior of Bayesian and SVM algorithms is worse than kNN.

On this point we discard work with other algorithm instead of kNN algorithm.

4.5. Normalization data

Thinking in the separation between the data measurement in the different devices we tried to work with margins instead of concrete power points. The way to do this is normalize all data-power measures. The normalization had been between 0 and -1.

We use Weka software to train and test the different devices using KNN algorithm with K=1.

The first test analysis consists in training and testing with the same device. The result shows on Table 12 for three devices.

Evaluation on training set	MacBook		IPhone 5S		Windows-PC	
Correctly Classified Instances	1367	75.944%	1450	80.556%	1352	75.111%
Incorrectly Classified Instances	433	24.056%	350	19.444%	448	24.889%
Kappa statistic	0.7526		0.8		0.744	
Mean absolute error	0.017		0.0139		0.0178	
Root mean squared error	0.092		0.083		0.0938	
Relative absolute error		31.505%		25.695%		32.911%
Root relative squared error		55.996%		50.489%		57.102%
Total Number of Instances	1800		1800		1800	

Table 12. Summary data normalization evaluation on training set

Comparing the results obtained on Table 12 with the Table 6 on point 5.1 is possible distinguished a worst behavior. In concrete around 2% worst.

In the same route, we analyze the behavior training with a device and testing with other one. The Table 13 shows de results training map by IPhone 5S and testing MacBook and Windows PC.

Training by IPhone 5s	MacBook		Windows-PC	
Correctly Classified Instances	95	5.2778%	30	1.667%
Incorrectly Classified Instances	1705	94.722%	1770	98.333%
Kappa statistic	0.0257		-0.0114	
Mean absolute error	0.0534		0.0546	
Root mean squared error	0.2179		0.2238	
Relative absolute error		98.955%		101.079%
Root relative squared error		132.609%		136.156%
Total Number of Instances	1800		1800	

Table 13. Summary on training by IPhone 5S

The Table 14 shows de results training map by MacBook and testing IPhone 5S and Windows PC.

Training by MacBook	IPhone 5S		Windows-PC	
Correctly Classified Instances	125	6.944%	62	3.444%
Incorrectly Classified Instances	1675	93.056%	1738	96.556%
Kappa statistic	0.0429		0.0069	
Mean absolute error	0.0518		0.0536	
Root mean squared error	0.222		0.2227	
Relative absolute error		95.813%		99%
Root relative squared error		135.092%		136%
Total Number of Instances	1800		1800	

Table 14. Summary on training by MacBook

The results obtained on Table 13 and Table 14 are worst like happens on the first test.

The behavior of the data normalize is worse than without. The results had not been like we expected. In this case we expected improve the behavior working with the margin instead of the power value.

At this point we can claim that the technology does not work when we use a device-data for training and a different device for testing. According with this situation, we continue analyze the behavior only in the case where training and testing are do it by the same device.

4.6. True positive analysis

Weka takes into account measurements such as True Positives (TP). A TP [10] is defined as the proportion of measurements labeled as class x which truly belong to class x out of all measurements made. Therefore, it defines the success rate on determining the measurement's "real" location.

		0	1	2	3	4	5	6	7	8
MacBook	Zone 0	0.88	0.1	0.88	0.84	0.94	0.72	0.78	0.66	0.92
	Zone 1	0.94	0.34	0.4	1	0.36	0.86	0.56	0.98	0.96
	Zone 2	0.82	1	0.96	1	0.82	0.76	0.88	0.74	0.18
	Zone 3	1	0.8	0.78	1	0.74	0.96	0.9	0.8	0.96

Figure 20. MacBook heat graph of TP

The Figure 20 shows a heat distribution of these true positives for the MacBook case (all heat graph are available at Appendix D).

4.7. Surface room's effect

On this point we determined if the surface of room affects the power distribution or if it is possible to find a pattern that repeats for all devices.

We used Weka software simulation, applying the kNN algorithm and using the same device for training and testing.

We show the true positive (TP) results. This is a value between 0-1. When the value is 1 this means 100% success, when it is 0.5 this means 50% success and so on.

		0	1	2	3	4	5	6	7	8
Zone 0	MacBook	0.88	0.1	0.88	0.84	0.94	0.72	0.78	0.66	0.92
	Windows-PC	0.8	0.9	0.28	0.92	1	0.88	0.86	0.8	0.96
	IPhone	0.9	0.98	0.86	0.32	0.34	0.6	0.7	0.64	0.36
Zone 1	MacBook	0.94	0.34	0.4	1	0.36	0.86	0.56	0.98	0.96
	Windows-PC	0.62	0.72	0.46	1	0.92	0.5	1	0.82	1
	IPhone	0.16	1	0.94	0.92	0.78	0.8	0.92	0.84	0.8
Zone 2	MacBook	0.82	1	0.96	1	0.82	0.76	0.88	0.74	0.18
	Windows-PC	0.88	0.58	0.76	0.32	0.78	1	0.58	1	0.94
	IPhone	1	0.94	0.92	0.9	0.94	0.92	0.8	1	0.96
Zone 3	MacBook	1	0.8	0.78	1	0.74	0.96	0.9	0.8	0.96
	Windows-PC	0.76	0.58	0.82	0.48	0.82	0.88	0.66	1	0.64
	IPhone	1	0.86	0.98	0.8	0.96	0.92	0.82	0.98	1

Figure 21. Comparison graph areas

The Figure 21 shows the true positive results for the four zones delimited on the room 130. We use the heat graph to compare the behavior in a concrete are by the different devices. As shows on zone 0 point 1, for the MacBook the true positive is 10% of the test, but on the Windows and on the IPhone is around 90-100%.

If we open the span and focused on the behaviors below 70% we can see more errors in the zone 0 and Zone 1 than the other two zones. But it is not sufficient to determine, which may have an influence affecting the surface will occur, in the same way, the readings of the three devices.

According to the results it is not possible to establish any link between the true positive and the surface-room. So, we can conclude that the behavior of the Wi-Fi signal does not been affected by the room design.

4.8. Changing data collection area

An objective of this thesis is test the ability to locate a person using Wi-Fi signals in a restaurant, at a table in a concrete and specific seat in order to develop an intelligent management system for restaurants.

Taking in consideration the results obtained, the Wi-Fi signals it is not enough to have this kind of precision.

On this point we investigate over the same room the precision point to distinguish between devices. For do this, we increase the separation between point-measures following different patterns areas:

- 1.5 m^2 in a square area.
- 3 m^2 in a square area.
- 6 m^2 in a square area.
- Chess combination.

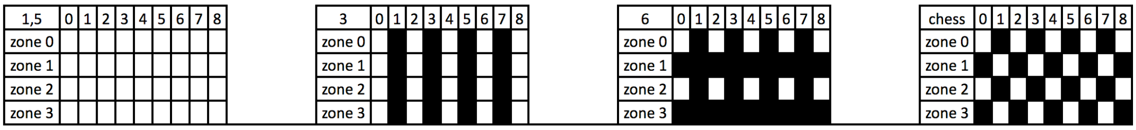


Figure 22. Data pattern distribution

Figure 22 shows the different patterns where the white areas are power data to process and the black areas represents the power data eliminated.

To present the results we use a graph to distinguish success areas and critical areas. It only presents the IPhone 5S case. The other results appear on Appendix D.

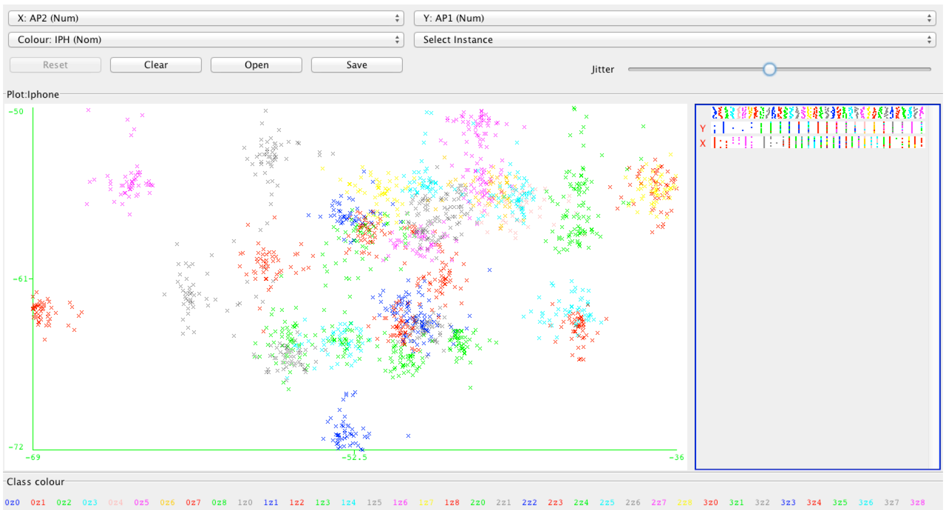


Figure 23. Power Training distribution Iphone5S 1.5m2

At this point we try to define a new value for the area. Like we show on the Figure 23 using the initial precision areas is not possible determine a concrete point because exist a big concentration of signals in the same zone. For this reason now we will work with different areas starting on 3m^2, trying to use the dispersion areas were is easier make localization.

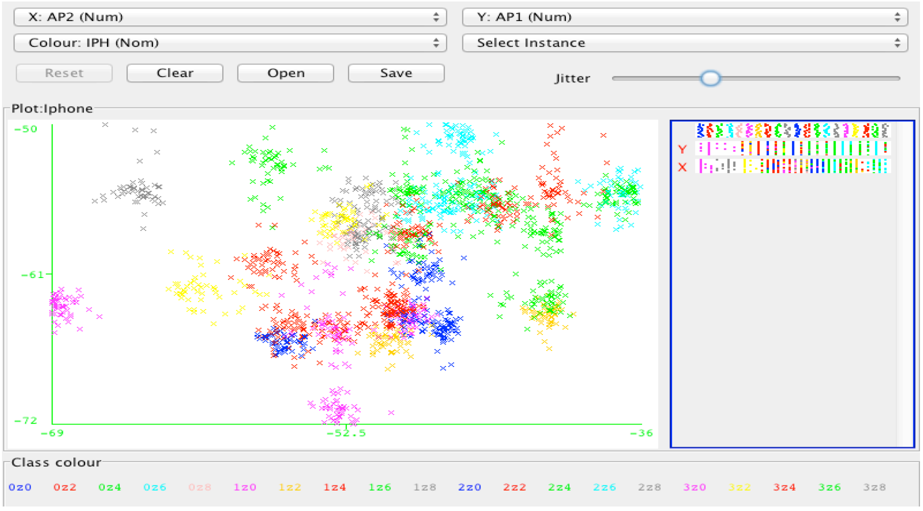


Figure 24. Power Training distribution IPhone 5S 3m2

When we increase the area the concentration points are reduced like shows on the Figure 24. In this case we are working with an area of 3m2. On the Windows-PC, the cloud points continue been bigger. For this reason it is not possible identify different areas, this device being the most critical. Now we test an area of 6m2.

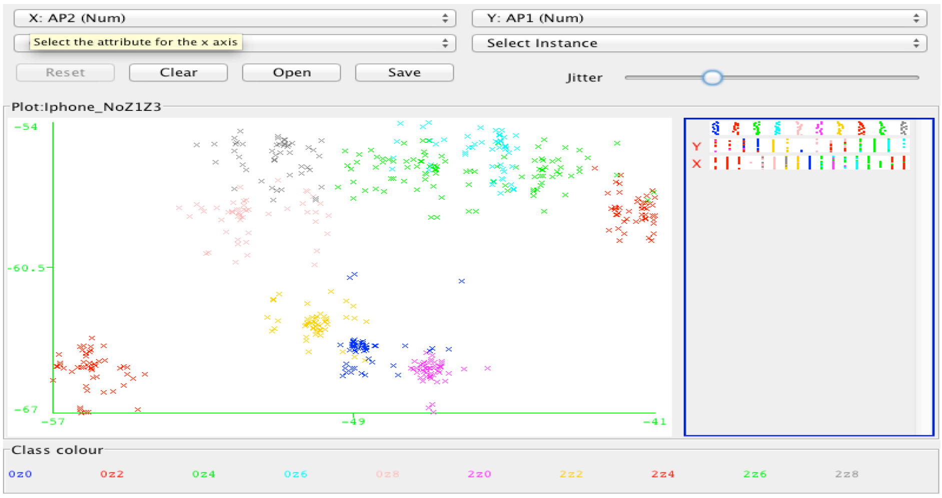


Figure 25. Power training distribution IPhone 6m2

When we increase the precision areas improves the recognition surface. We show this reality on the Figure 25.

At this moment we experiment with other combination. Taking in consideration the mapping area we eliminate the measurements of some areas. To determine which areas we will eliminate we follow a structure like ‘chess’.

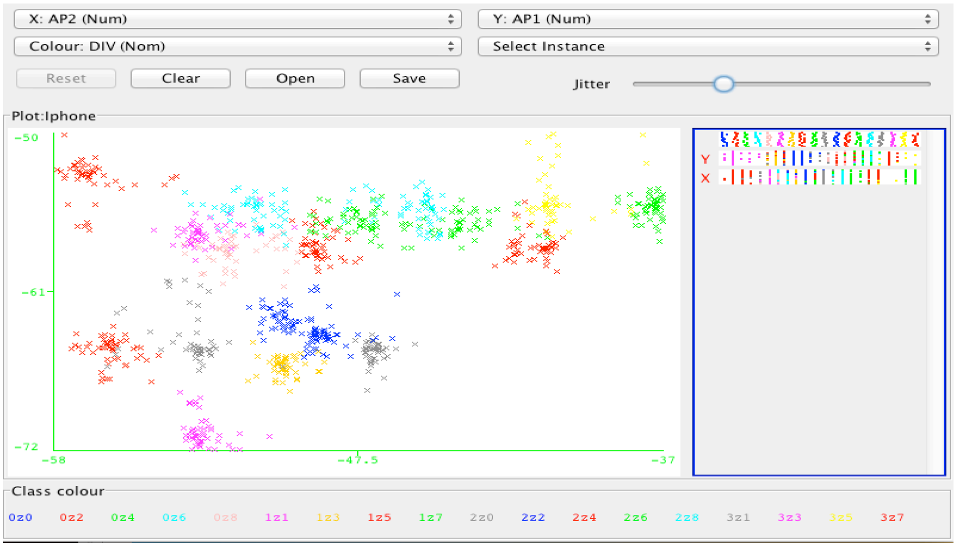


Figure 26. Power training distribution IPhone Chess

The behavior using this pattern is really good as shown on Figure 26. At the end is a surface of 3m2 but if compares with the graph on Figure 24 the cloud points are minimized.

The Table 15 resumes the behavior modifying the coverage area.

	IPhone 5S							
Comparative on training set	1.5m2		3m2		6m2		Chess	
Correctly Instances	1478	82%	837	84%	477	95%	845	94%
Incorrectly Instances	322	17%	163	16%	23	5%	55	6%
Kappa statistic	0.816		0.8284		0.9489		0.9353	
Mean absolute error	0.0129		0.0207		0.0144		0.0103	
Root mean squared error	0.08		0.1011		0.0832		0.0707	
Relative absolute error		23%		22%		8%		10%
Root relative squared error		48%		46%		28%		31%
Total Number of Instances	1800		1000		500		900	

Table 15. Comparative on training set for IPhone

Like shows the Table 15 the best pattern is the separation presents by the chess case. The other devices present the same characteristics as shown on Appendix D.

CHAPTER 5. Environmental impact

The purpose of the assessment is to ensure that decision makers consider the environmental impacts when deciding whether or not to proceed with a project

This project is based on software tools. For this purpose, various devices have been used. The environmental impact of this project is transferred to the level of devices used. The devices used are MacBook Pro Retina, IPhone 5S y PC Acer Aspire One.

Therefore, three points of consideration are presented according with the devices.

5.1. Climate Change

Greenhouse gas emissions have an impact on the planet's balance of land, ocean, and air temperatures. Most of device's corporate greenhouse gas emissions come from the production, transport, use, and recycling of its products. Apple and Acer seek to minimize greenhouse gas emissions by setting stringent design-related goals for material and energy efficiency.

The chart below provides the estimated greenhouse gas emissions for the 13-inch MacBook Pro with Retina display over its life cycle [16]

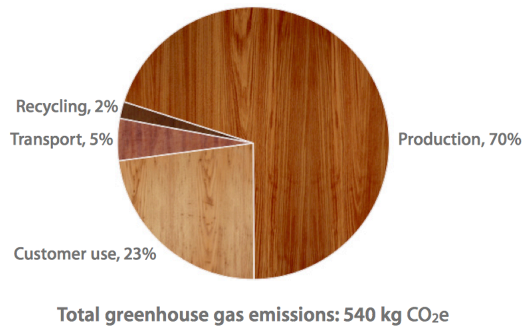


Figure 27. Greenhouse Gas Emissions for MacBook Pro[16]

5.2. Energy Efficiency

Because one of the largest portions of product-related greenhouse gas emissions results from actual use, energy efficiency is a key part of each product's design. Apple and Acer products use power- efficient components and software that intelligently powers them down during periods of inactivity.

5.3. Material Efficiency

Device's ultra-compact product and packaging designs lead the industry in material efficiency. Reducing the material footprint of a product helps maximize shipping efficiency. It also helps reduce the energy consumed during production and material waste generated at the end of the product's life. Waste is further minimized through the use of batteries that last up to three times longer than typical notebook batteries.

The IPhone 5S with Retina [16] display is made of aluminium and other materials highly desired by recyclers. The chart below details the materials used in this model.

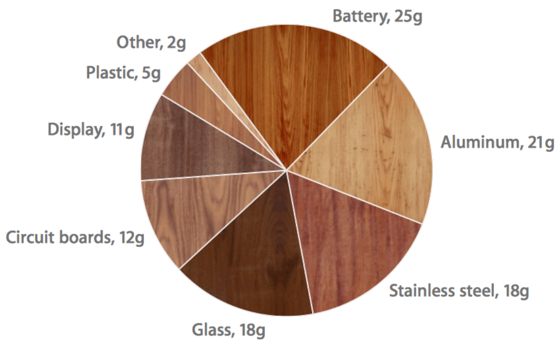


Figure 28. Material use for IPhone 5S

CHAPTER 6. Conclusions and further work

This Thesis presents Qpides! as an App for restaurants and it is mainly focused in the location part of the application, in particular, the location of a person in a concrete chair in a specific table inside a restaurant. Qpides! works on the idea of optimizing the indoor localization based on Wi-Fi signals using a free analyzing tool.

This Thesis presents (i) an overview regarding the different ways of indoor location and (ii) identifies three solutions (i.e., Google indoor, Insiteo and RedPin). We conducted an exhaustive analysis of the application RedPin to understand their behavior on the context of this Thesis. Also, this analysis helps us for working with our fingerprint project.

To properly evaluate the data, we created a simulation of a restaurant on the room 130 of the UPC campus, as it was a real restaurant scenario. We made the mapping area using different user devices and the room's access-points. Also, we used Weka for training and testing our data-map.

After considering and analyzing the three possible solutions on Chapter 2 (i.e. Google indoor, Insiteo and RedPin), there are some questions that remain to be clarified, such as, what happens when users enter incorrect fingerprints, either consciously or unconsciously, or how to treat when we have ambiguous labels

6.1. Conclusions

According to the results obtained on the evaluation chapter we can conclude that:

(i) The data obtained, with different devices for Wi-Fi signals, does not have a common behavior.

(ii) The kNN algorithm has better perform than the other algorithm analyzed.

(iii) The k value equal to 1 for the kNN algorithm presents better behavior than other k values.

(iv) The fingerprint works successful when training the map with the same device used for test it, but when the devices are mixed they don't work properly.

(v) When we increase the separation between power measurements we improve the results of our location-test, but we lose precision.

(vi) In accordance with the conclusion (v) above, where we mentioned the separation measurements, we conclude that working with the power-measurement pattern is possible improving the allocation behavior without losing precision.

Taking into account the evaluation conclusions we can assert that:

(i) The behavior of Wi-Fi signals change in indoor dynamic environment depending on various unpredictable factors. RSSI distribution might substantially change over different periods, duration, and frequency, invalidating the assumption that data distribution does not change much over time. The presence of people and their continuous movement affect Wi-Fi behavior in unpredictable ways, which makes indoor environments even unstable. We also showed that filtering techniques based on AP visibility might not work due to changes in APs visibility between measurement periods of different lengths.

(ii) The Wi-Fi signal can be used to identify (i) in which location/room is the device (for example, in which restaurant it is), and (ii) once we have done the aforementioned, and complementing the Wi-Fi signal with other technologies, where exactly is located the device (for example, in which table it is).

(iii) Using fingerprint technology, with Wi-Fi signals only, does not bring a solution for the location goal of the Qpides! App. It is necessary to use a fingerprint technology with the combinations of different signals, as Bluetooth or GSM, to improve and make a good accuracy positioning.

According to my own experience doing this Thesis, I conclude that:

(i) It was interesting working on this research and I am happy with the subject that I chose because this Thesis has allowed me to work on (1) researching, (2) analyzing large sets of data, (3) developing applications for iPhone, and (4) studying the user's reaction.

(ii) It was also a great experience to learn a new programming language (i.e. Objective-C), and developing applications for iPhone and Mac OS X.

(iii) I learned a lot about indoor localization, which is a wide and challenging topic with a lot to research and understand. Also, it was interesting to study the behavior of Wi-Fi signals and to realize just how unpredictable they can be. I found it especially intriguing to study many factors, which effect Wi-Fi signals, and how dynamic and constantly changing they are. Finally, it was challenging working with a large collection of fingerprints, analyzing them and picking measurement intervals of various lengths.

6.2. Additional work to do

During the analysis of the data for this Thesis, I realized that there are additional ways to try to improve the localization inside a restaurant using Wi-Fi signals only that may be interesting to further analyze but they were outside of the scope of this Thesis. These ways are described below:

(i) Using a reference signal in which the server could adapt the fingerprint to each user. This is like applying a correction to the readings obtained by the user's device.

(ii) Having a large number of fingerprints on the server so, once the user enters in the restaurant, his device would send the readings of the access points and the server would select the fingerprint, which is more suitable for the user based on the similarity of data.

(iii) Looking location algorithms, analyze whether the combination of two or more algorithms would bring better accuracy.

(iv) Regarding power, measure the distance between two APs, and with this information, try to adapt the fingerprint for all devices.

References

[1] Bolliger, P., RedPin, 2008. Software available at http://www.RedPin.org/

[2] P Bolliger. RedPin-adaptive, zero-conjuration indoor localization through user collaboration. . . . workshop on Mobile entity localization and tracking in . . . , 2008. URL http://portal.acm.org/citation.cfm?id=1410025.

[3] Mobile phones URL http://xombit.com/2013/10/smartphones-mas-vendidos-espana

[4] Li, B., Salter, J., Dempster, A.G., and Rizos, C.,"Indoor positioning techniques based on Wireless LAN" IEEE Int. Conf., (Mar. 2006).

[5] Brunato, M., and Battiti, R., "Statistical Learning Theory for Location Fingerprinting in Wireless LANs", Computer Networks, (2005).

[6] Google Maps; indoor mapping. URL http://www.google.com/intl/es-419/maps/about/partners/indoormaps/

[7] AirRadar. URL www.koingosw.com/products/airradar.php

[8] Theodoridis S. Koutroumbas K. *Pattern Recognition*, Second Edition, pp. 44-50, Academic Press (2003-04-09)

[9] Kappa URL http://en.wikipedia.org/wiki/Cohen%27s_kappa

[10] Weka URL http://weka.wikispaces.com/Primer

[11] Word reference URL http://www.wordreference.com

[12] K. Kaemarungsi. ,"Design of indoor positioning systems based on location _fingerprinting technique". University of Pittsburgh, Jan 2005.

[13] P. Bolliger, K. Partridge, M. Chu, M. Langheinrich, "Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling". Palo Alto Research Center, Palo Alto, CA, USA, Academic Press (2011)

[14] B. Sikdar, "A Study of the Environmental Impact of Wired and Wireless Local Area Network Access" Senior Member, IEEE

[15] M. Havas "Analysis of Health and Environmental Effects of Proposed San Francisco Earthlink Wi-Fi Network"; Trent University, Peterborough, Canada , May 2007

[16] Apple URL: https://www.apple.com/environment/reports

[17] Acer URL: http://us.acer.com/ac/en/US/press/2014/75490

[18] Indoor Location startups innovating Indoor Positioning. URL: http://dondodge.typepad.com/the_next_big_thing/2013/06/indoor-location-startups-innovating-indoor-positioning.html

ACRONYMS

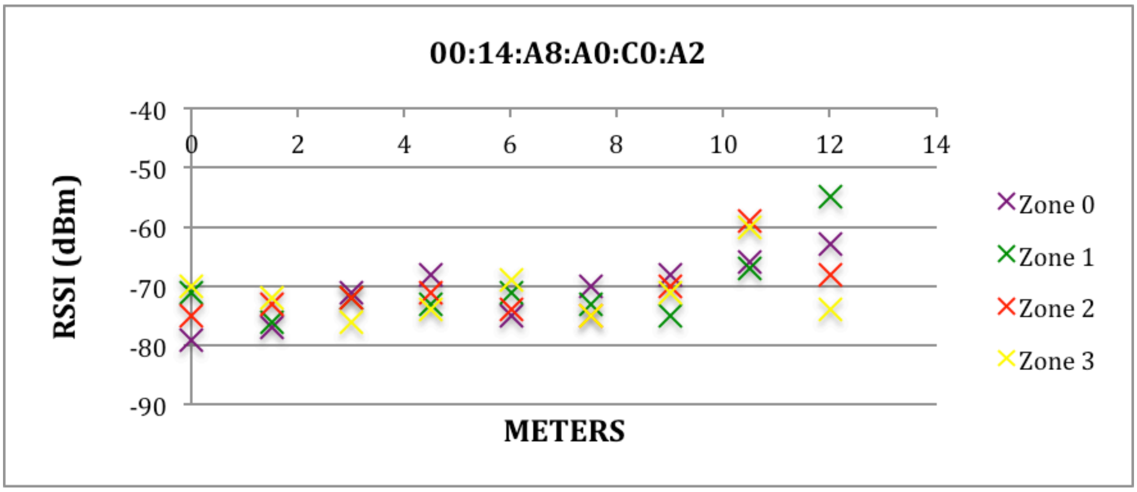
AP	Access point
App	Application
BSSID	Basic service set identifier
BD ADDR	Bluetooth device address
CI	Cell identifier
GSM	Global system mobile
LAC	Location area code
MAC	Media access control
MCC	Mobile country code
MNC	Mobile network code
NN	Nearest Neighbor
RSS	Received signal strengths
RSSI	Received signal strengths indicator
SSID	Service set identifier
SVM	Support Vector Machines
SVO	Sequential Minimal Optimization
UMTS	Universal mobile telecommunications system
TP	True positives

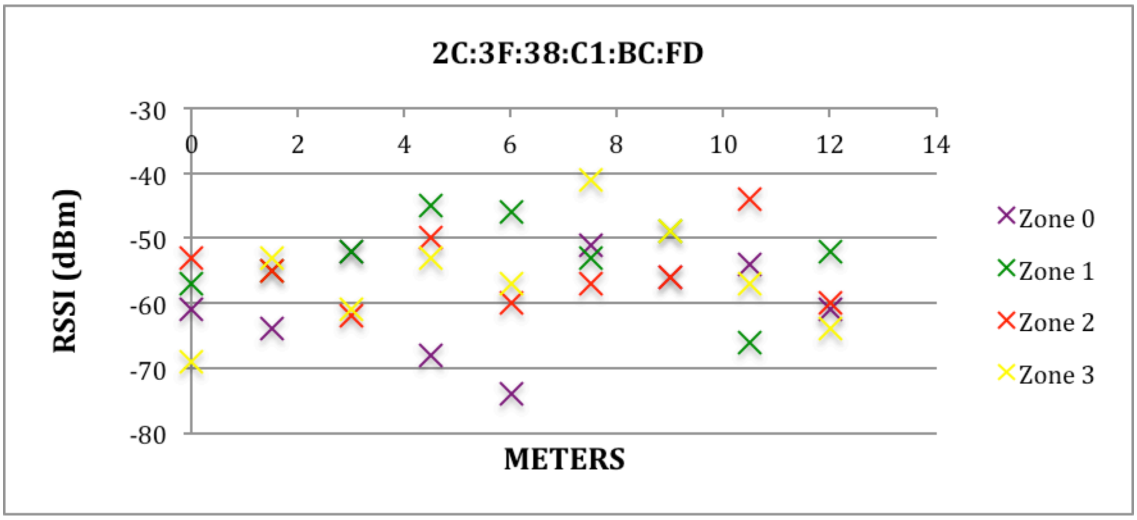
APPENDIX A. POWER DISTRIBUTION

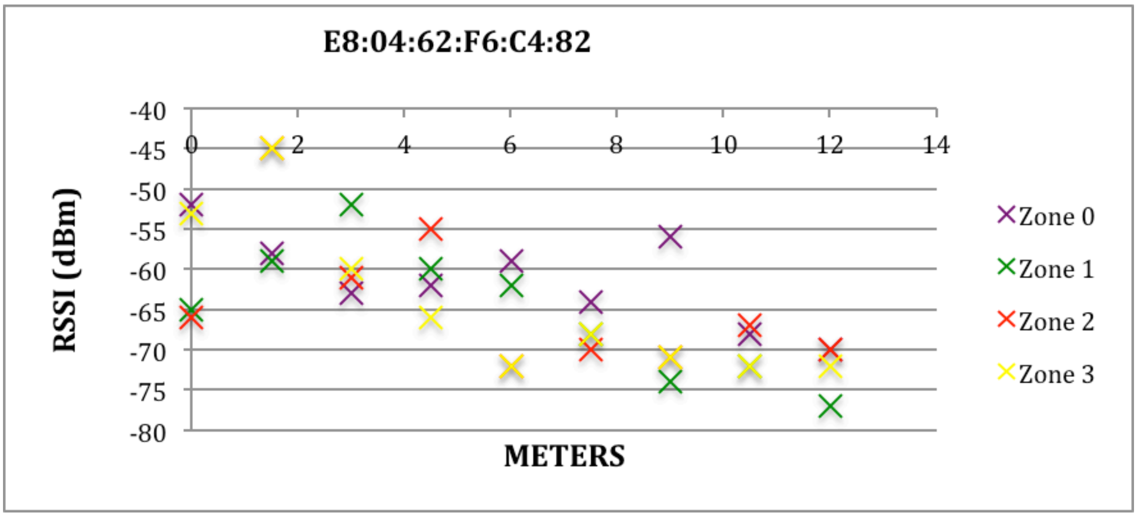
Windows-PC power distribution

For Windows-PC it used Software called inSSIDer. The results are presented for the different zones and it identified the three BSSID that we analyze.

Now it presents the mediana-power result in a graph to make ease the fingerprint elaboration. And it helps to know the behavior of the each BSSID on the room.



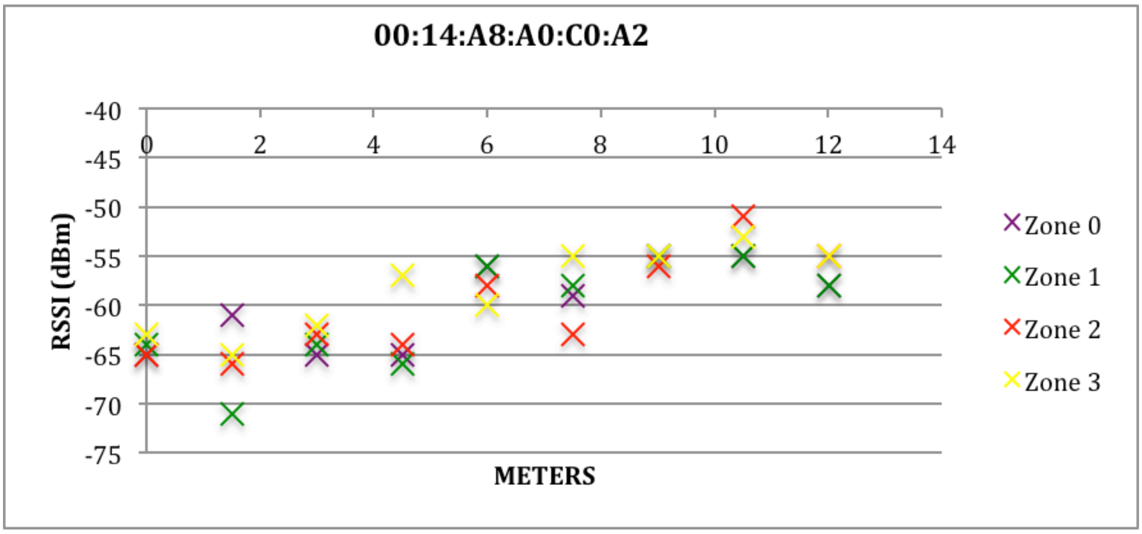


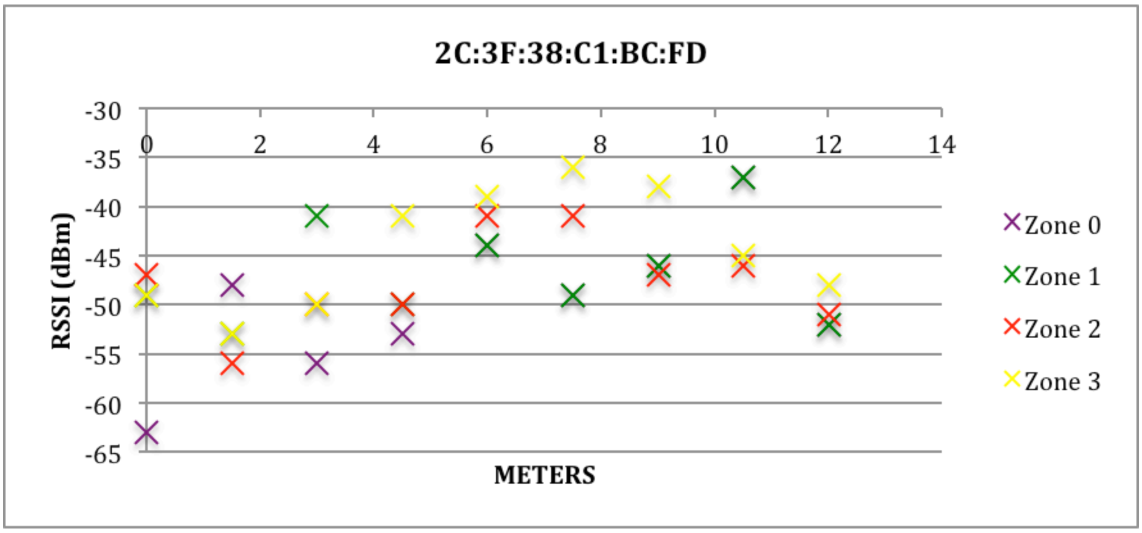


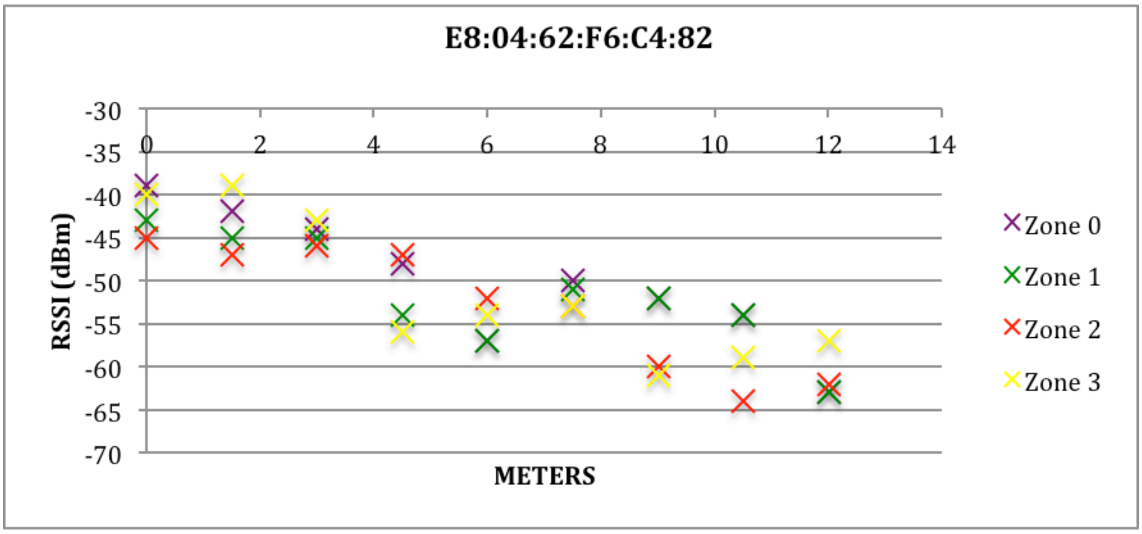
IPhone 5S power distribution

For the phone it used Software called WifiScaner. The results are presented for the different zones and it identified the three BSSID that we analyze.

Now it presents the mediana-power result in a graph to make ease the fingerprint elaboration. And it help to know the behavior of the each BSSID on the room.



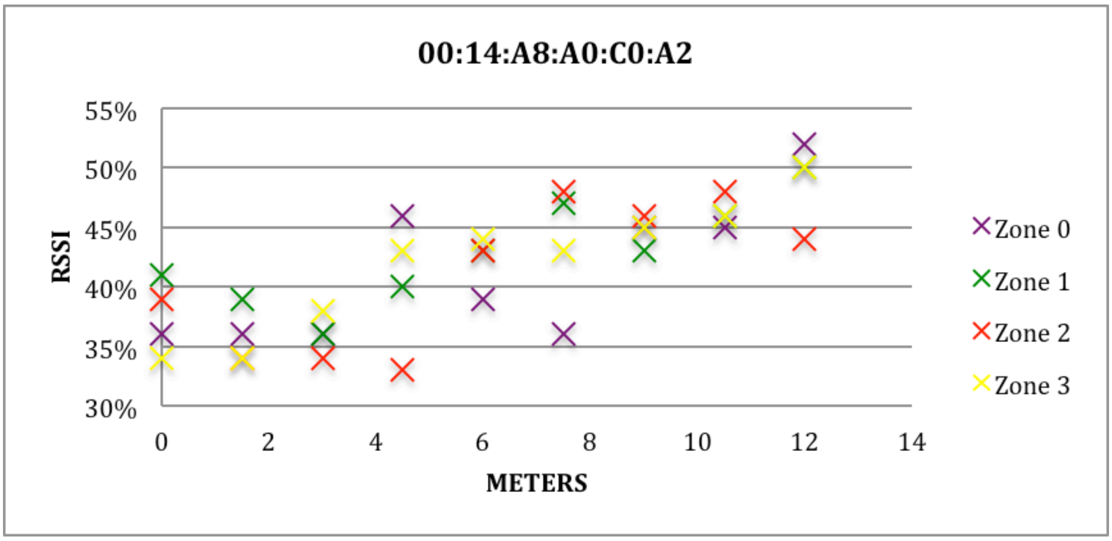


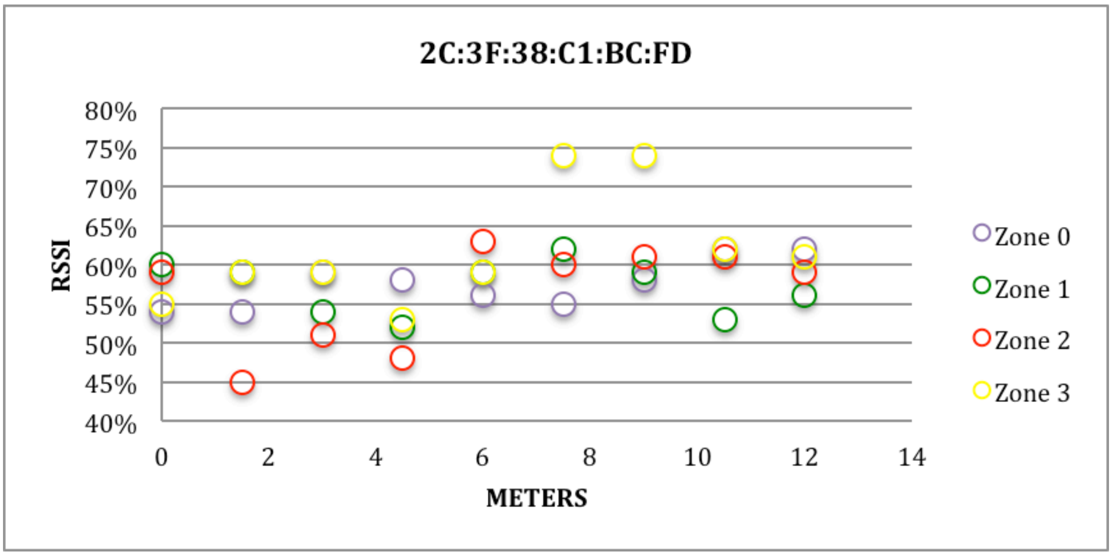


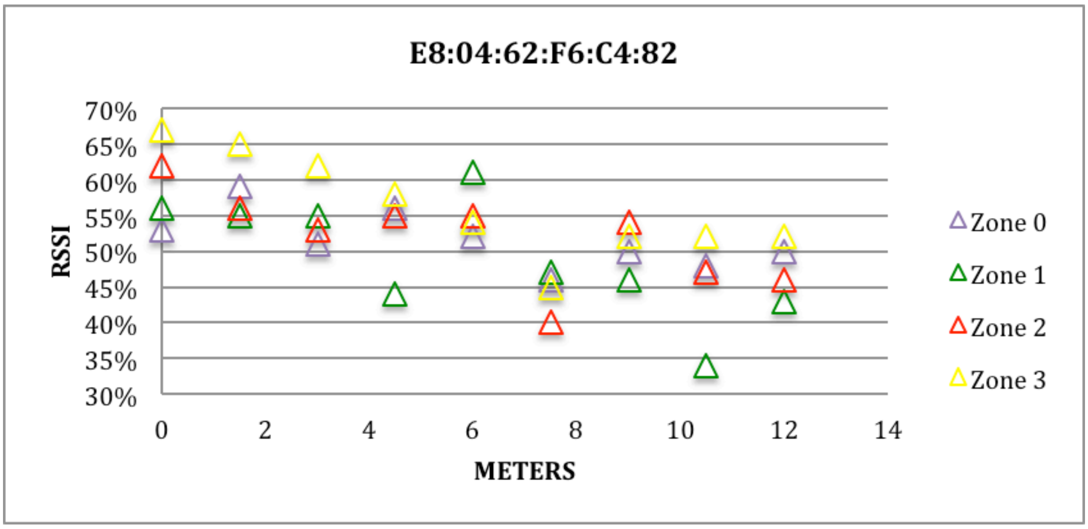
MacBook pro power distribution

For Mac-PC it used Software called AirRadar. The results are presented for the different zones and it identified the three BSSID that we analyze.

Now it presents the media power result in a graph to make ease the fingerprint elaboration. And it helps to know the behavior of the each BSSID on the room.


00:14:A8:A0:C0:A2
RSSI
55%
50%
45%
40%
35%
30%
0
2
4
6
8
10
12
14
METERS
Zone 0
Zone 1
Zone 2
Zone 3


2C:3F:38:C1:BC:FD
RSSI
80%
75%
70%
65%
60%
55%
50%
45%
40%
0
2
4
6
8
10
12
14
METERS
Zone 0
Zone 1
Zone 2
Zone 3

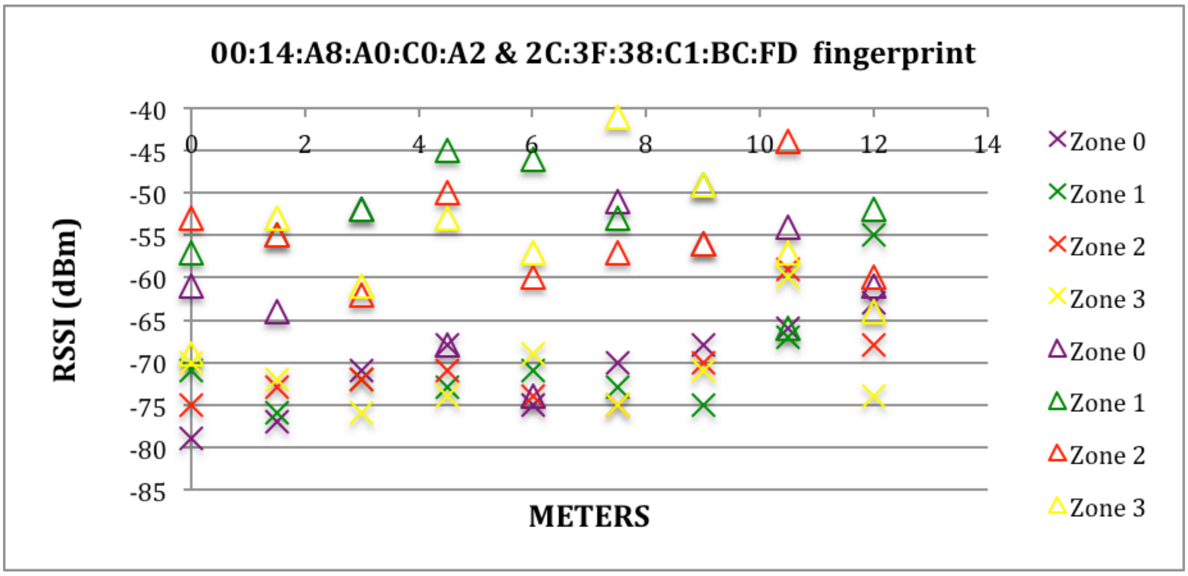

E8:04:62:F6:C4:82
RSSI
70%
65%
60%
55%
50%
45%
40%
35%
30%
0
2
4
6
8
10
12
14
METERS
Zone 0
Zone 1
Zone 2
Zone 3

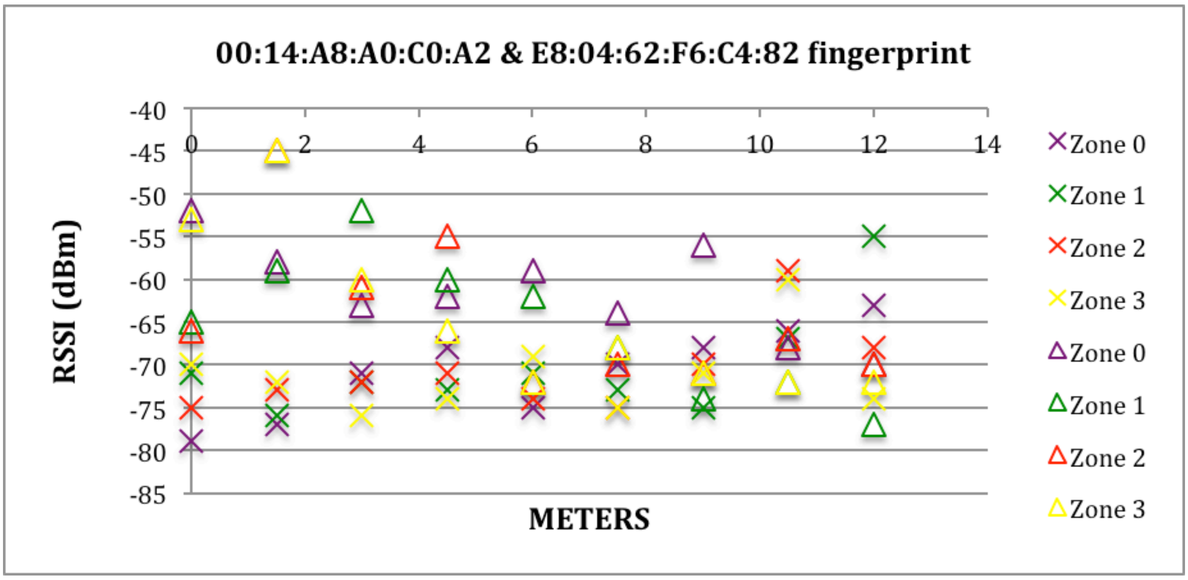
APPENDIX B. FINGERPRINT

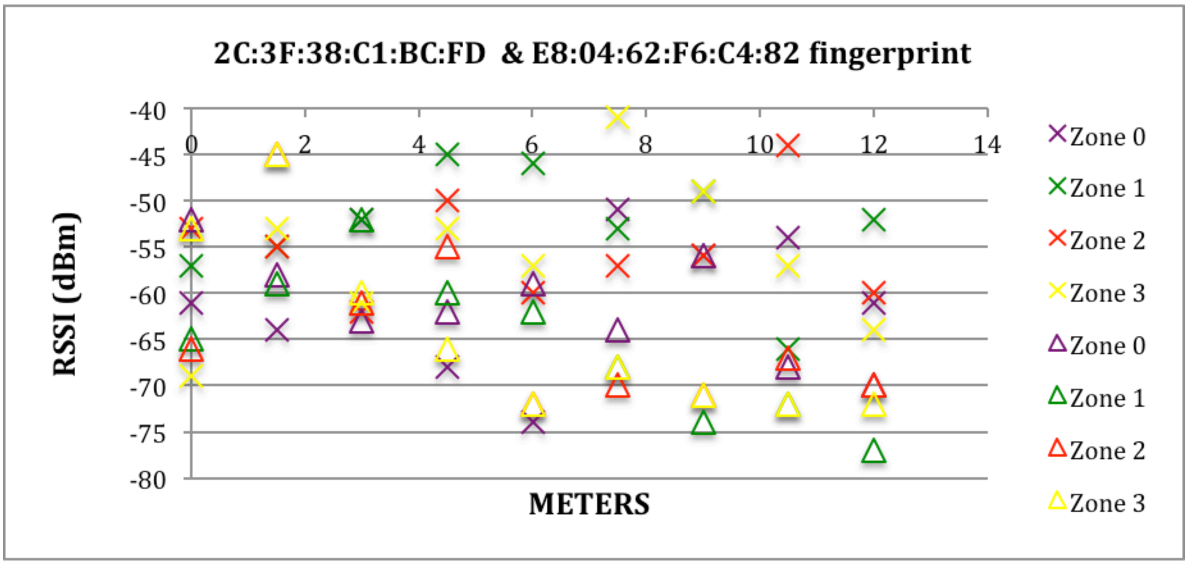
Now, taking in consideration the power distribution for the different BSSID, we combine in pairs the different BSSID to try to find a pattern depending on the RSSI. This pattern will be the fingerprint.

The main idea is identifier different power combination for the different areas.

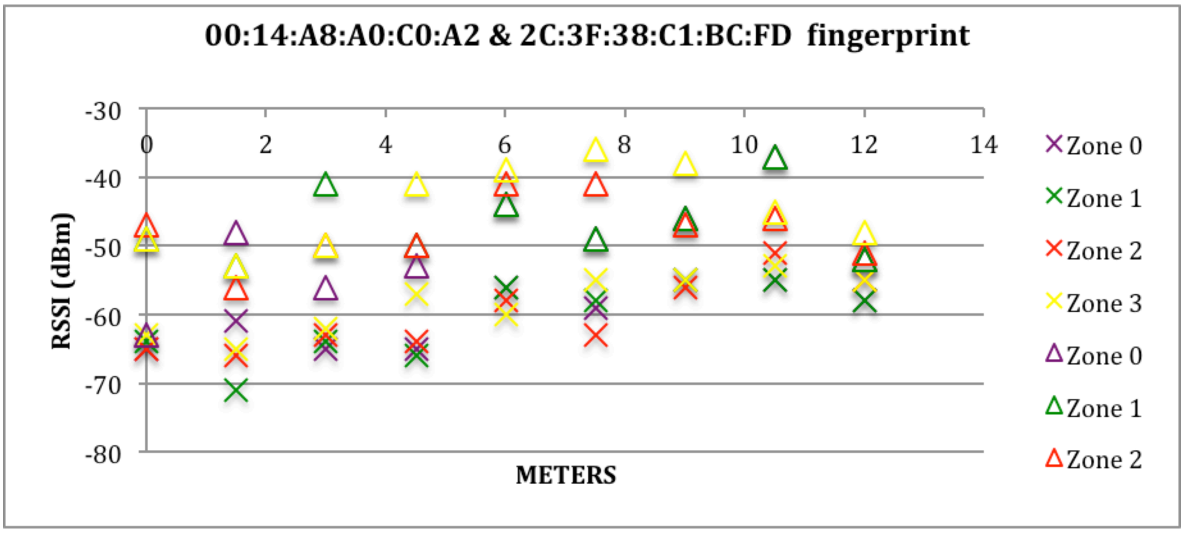
Windows-PC fingerprint

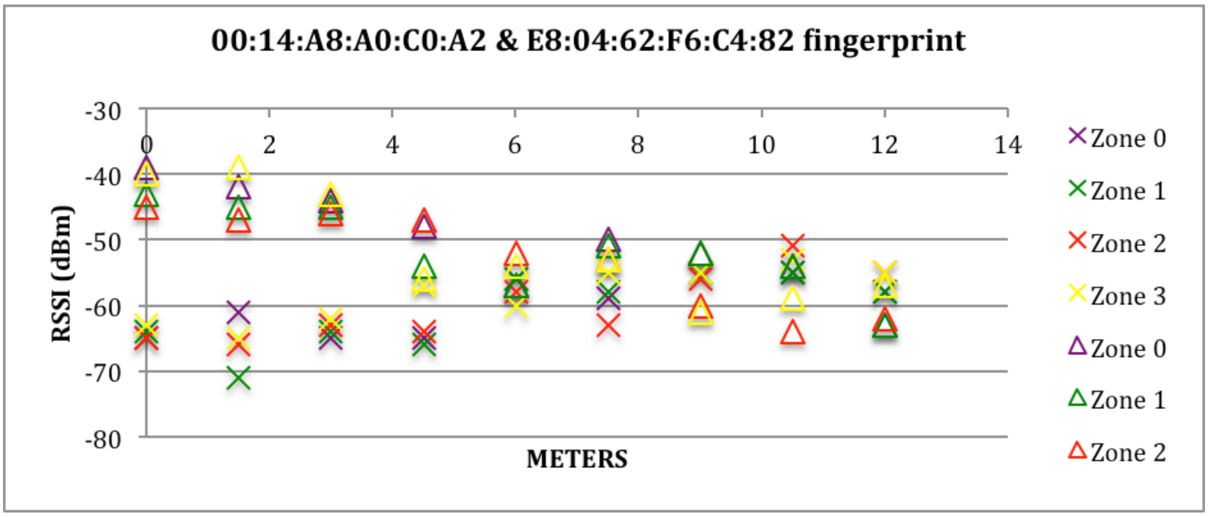


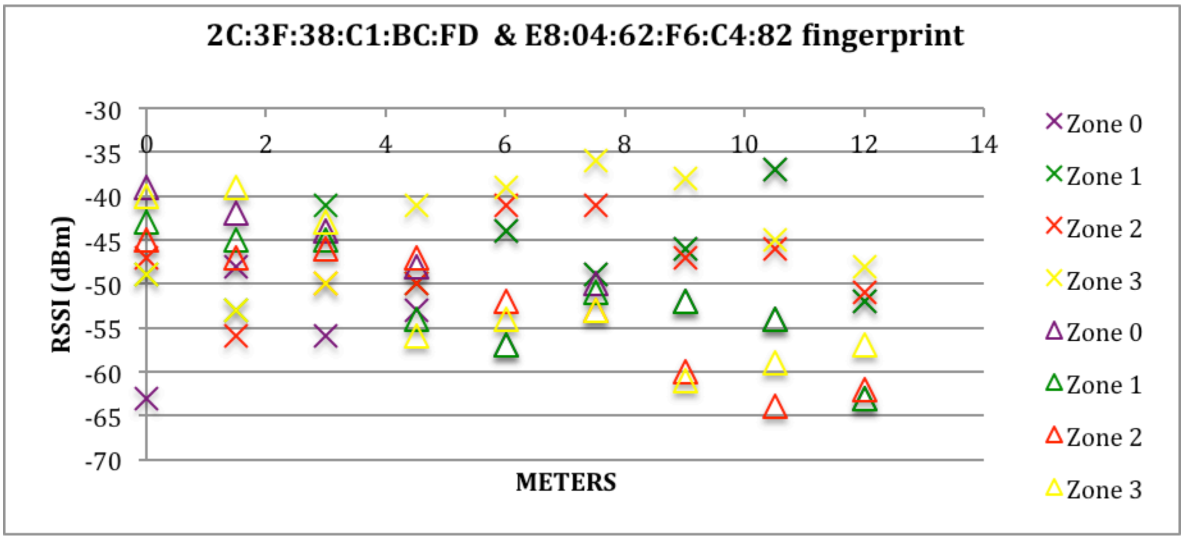




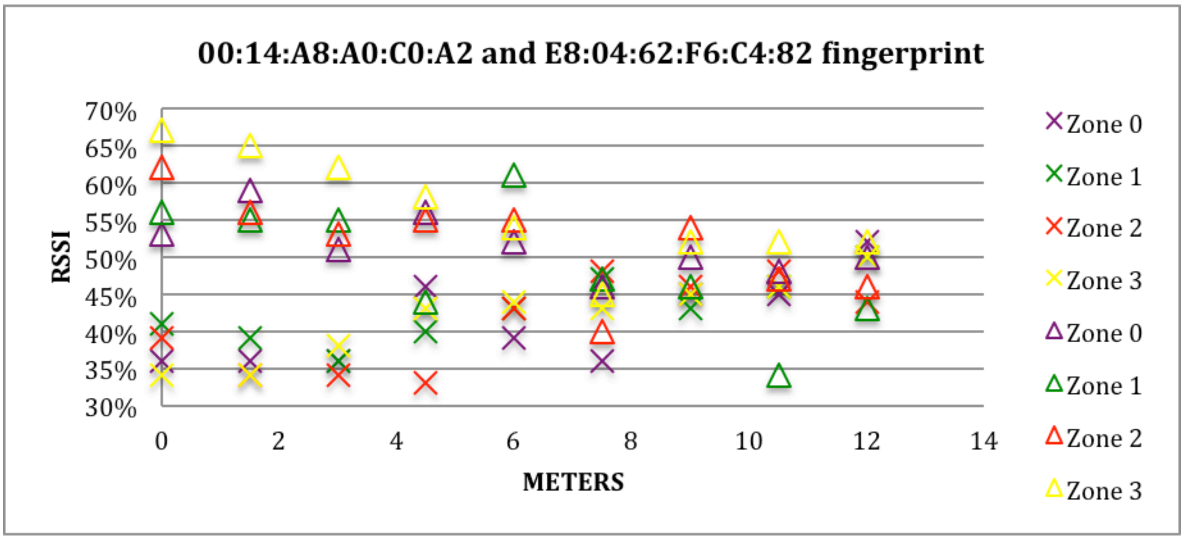
Iphone 5C fingerprint

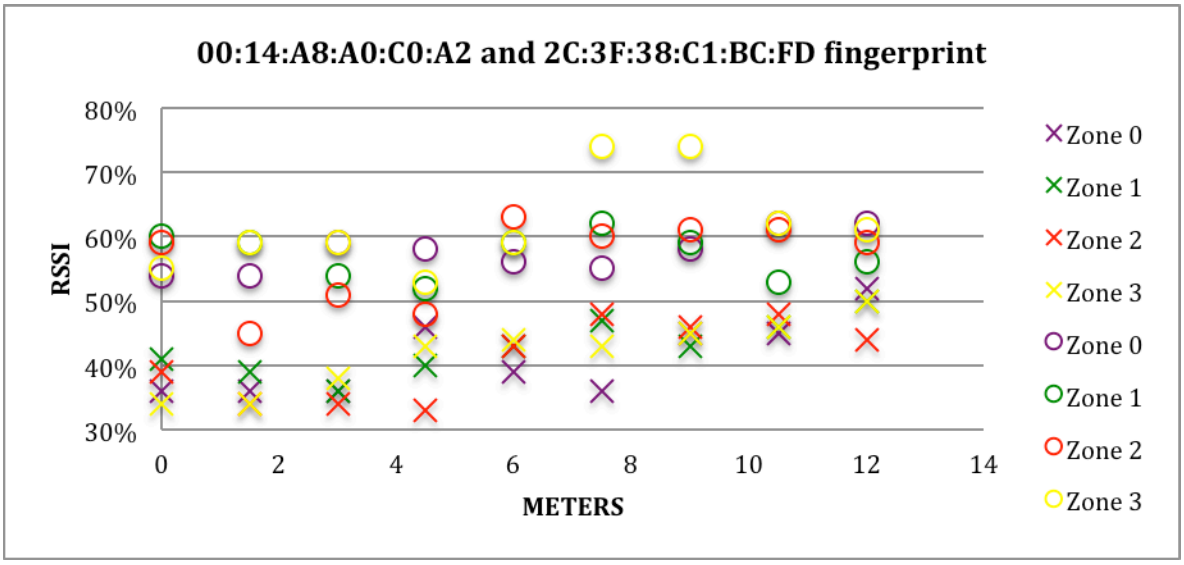


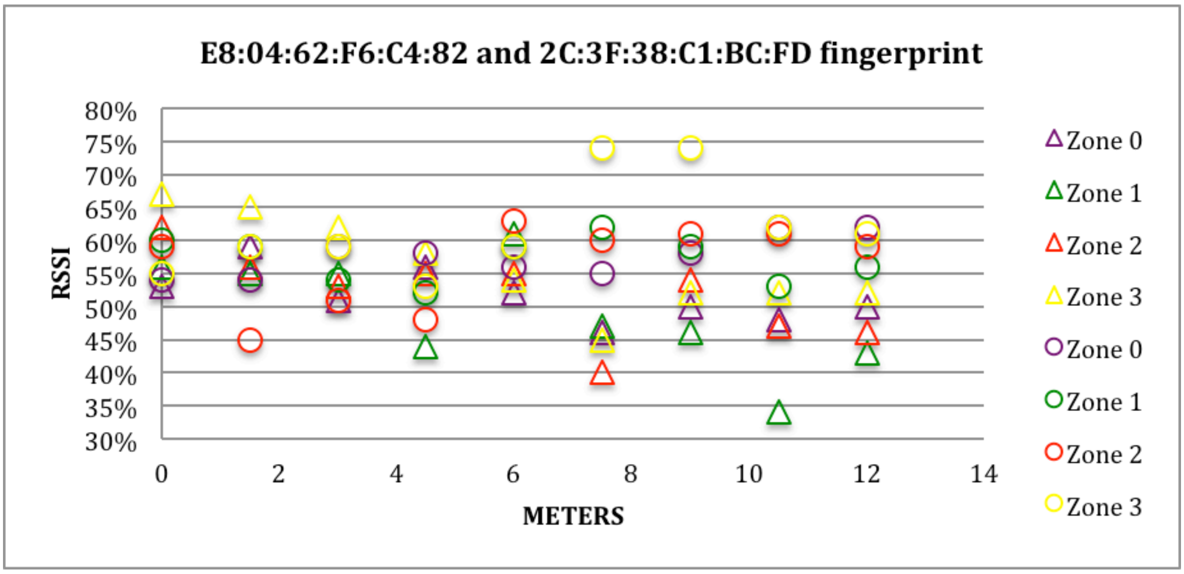




Macbook Pro fingerprint






E8:04:62:F6:C4:82 and 2C:3F:38:C1:BC:FD fingerprint
RSSI
80%
75%
70%
65%
60%
55%
50%
45%
40%
35%
30%
0
2
4
6
8
10
12
14
METERS
Zone 0
Zone 1
Zone 2
Zone 3
Zone 0
Zone 1
Zone 2
Zone 3

APPENDIX C. MEDIANA POWER CONTRIBUTION

Median power contribution space by combination of 00:14:A8:A0:C0:A2 and 2C:3F:38:C1:BC:FD access point.

Macbook-Pro

		0	1,5	3	4,5	6	7,5	9	10,5	12
Zone 0	00:14:A8:A0:C0:A2	-64	-64	-64	-54	-61	-64	-55	-55	-58
	2C:3F:38:C1:BC:FD	-46	-46	-41	-42	-44	-45	-42	-38	-38
Zone 1	00:14:A8:A0:C0:A2	-59	-61	-64	-60	-57	-53	-57	-54	-50
	2C:3F:38:C1:BC:FD	-40	-41	-46	-48	-41	-38	-41	-47	-44
Zone 2	00:14:A8:A0:C0:A2	-61	-66	-66	-67	-57	-52	-54	-52	-56
	2C:3F:38:C1:BC:FD	-41	-55	-49	-52	-37	-40	-39	-39	-41
Zone 3	00:14:A8:A0:C0:A2	-66	-66	-62	-57	-56	-57	-55	-54	-50
	2C:3F:38:C1:BC:FD	-45	-41	-41	-47	-41	-26	-26	-38	-39

Iphone 5S

		0	1,5	3	4,5	6	7,5	9	10,5	12
Zone 0	00:14:A8:A0:C0:A2	-65	-61	-65	-65	-56	-59	-55	-55	-58
	2C:3F:38:C1:BC:FD	-63	-48	-56	-53	-44	-49	-46	-37	-52
Zone 1	00:14:A8:A0:C0:A2	-64	-71	-64	-66	-56	-58	-55	-55	-58
	2C:3F:38:C1:BC:FD	-49	-53	-41	-50	-44	-49	-46	-37	-52
Zone 2	00:14:A8:A0:C0:A2	-65	-66	-63	-64	-58	-63	-56	-51	-55
	2C:3F:38:C1:BC:FD	-47	-56	-50	-50	-41	-41	-47	-46	-51
Zone 3	00:14:A8:A0:C0:A2	-63	-65	-62	-57	-60	-55	-55	-53	-55
	2C:3F:38:C1:BC:FD	-69	-53	-61	-53	-57	-41	-49	-57	-64

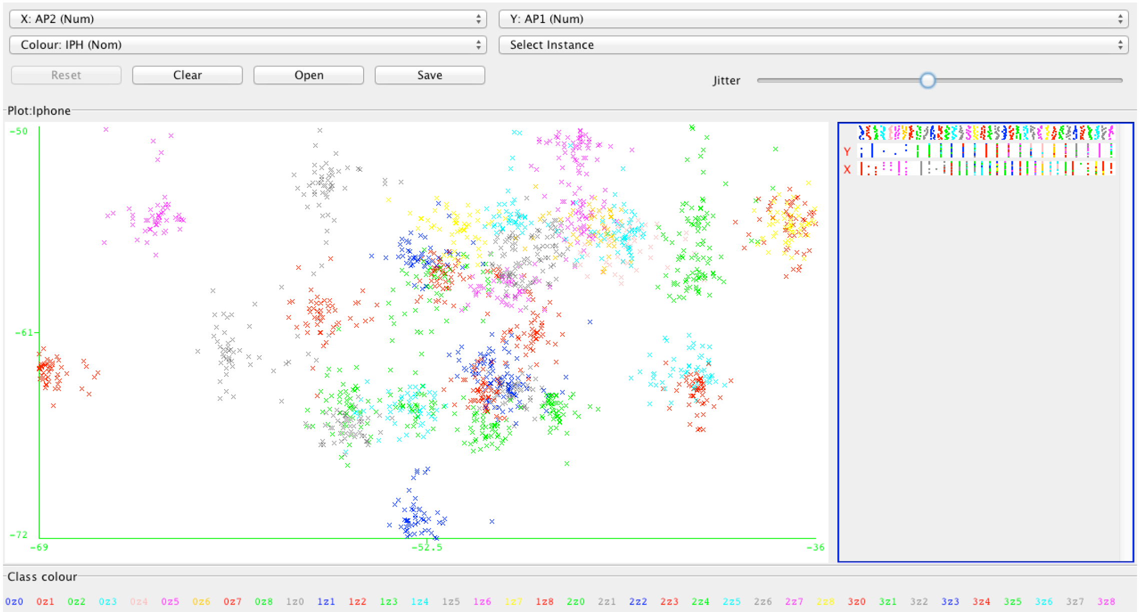
Windows-PC

		0	1,5	3	4,5	6	7,5	9	10,5	12
Zone 0	00:14:A8:A0:C0:A2	-79	-77	-71	-68	-75	-70	-68	-66	-63
	2C:3F:38:C1:BC:FD	-61	-64	-52	-68	-74	-51	-56	-64	-61
Zone 1	00:14:A8:A0:C0:A2	-71	-76	-72	-73	-71	-73	-75	-67	-55
	2C:3F:38:C1:BC:FD	-57	-55	-52	-45	-46	-53	-49	-66	-52
Zone 2	00:14:A8:A0:C0:A2	-75	-73	-72	-71	-74	-75	-70	-59	-68
	2C:3F:38:C1:BC:FD	-53	-55	-62	-50	-60	-57	-56	-44	-60
Zone 3	00:14:A8:A0:C0:A2	-70	-72	-76	-74	-69	-75	-71	-60	-74
	2C:3F:38:C1:BC:FD	-69	-53	-61	-53	-57	-41	-49	-57	-64

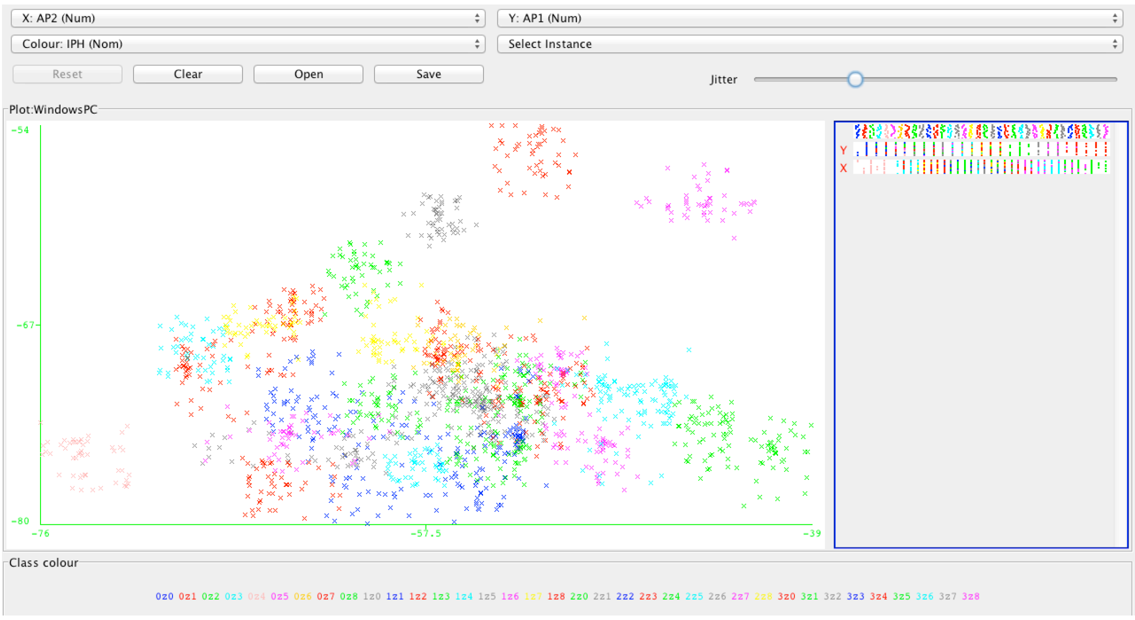
APPENDIX D. WEKA SIMULATION TEST

Plot area devices training

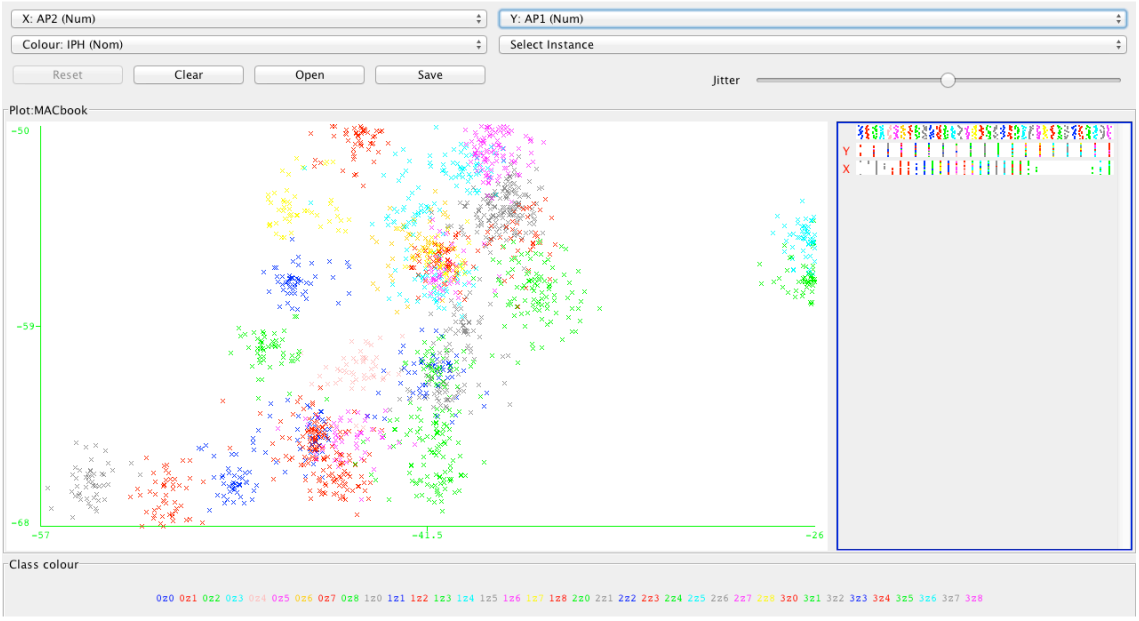
Power Training distribution IPhone 5S



Power training distribution Windows-PC



Power Training distribution MacbookPro



Test and training with same device results.

It be Training map used Macbook-pro signals and testing with the same device MacBook with k=1.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	1411	78.3889 %
Incorrectly Classified Instances	389	21.6111 %
Kappa statistic	0.7777 (0-1)	
Mean absolute error	0.0151	
Root mean squared error	0.0866	
Relative absolute error	27.915 %	
Root relative squared error	52.6801 %	
Total Number of Instances	1800	

It be Training map used WindowsPC signals and testing with the same device WindowsPC with k=1.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	1396	77.5556 %
Incorrectly Classified Instances	404	22.4444 %
Kappa statistic	0.7691	
Mean absolute error	0.0162	
Root mean squared error	0.0895	
Relative absolute error	29.9695 %	
Root relative squared error	54.446 %	

Total Number of Instances	1800

It be Training map used Iphone5s signals and testing with the same device Iphone5s with k=1.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	1478	82.1111 %
Incorrectly Classified Instances	322	17.8889 %
Kappa statistic	0.816	
Mean absolute error	0.0129	
Root mean squared error	0.08	
Relative absolute error	23.8862 %	
Root relative squared error	48.6526 %	
Total Number of Instances	1800	

Test and training with different device results.

It be Training map used Iphone5s signals and testing by MacBook with k=1.

k=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	99	5.5 %
Incorrectly Classified Instances	1701	94.5 %
Kappa statistic	0.028	
Mean absolute error	0.0527	
Root mean squared error	0.2221	
Relative absolute error	97.5304 %	
Root relative squared error	135.1781 %	
Total Number of Instances	1800	

It be Training map used Iphone5s signals and testing by Windows PC with k=1

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	103	5.7222 %
Incorrectly Classified Instances	1697	94.2778 %
Kappa statistic	0.0303	
Mean absolute error	0.0524	
Root mean squared error	0.2268	
Relative absolute error	96.9409 %	
Root relative squared error	138.0258 %	
Total Number of Instances	1800	

It be Training map used MacBook-pro signals and testing by IPhone 5s with K=1.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	91	5.0556 %
Incorrectly Classified Instances	1709	94.9444 %
Kappa statistic	0.0234	
Mean absolute error	0.0528	
Root mean squared error	0.2246	
Relative absolute error	97.6795 %	
Root relative squared error	136.66 %	
Total Number of Instances	1800	

It be Training map used Macbook-pro signals and testing by WindowsPC with K=1.

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          45               2.5    %
Incorrectly Classified Instances      1755              97.5    %
Kappa statistic                         -0.0029
Mean absolute error                      0.0542
Root mean squared error                  0.2296
Relative absolute error                100.2978 %
Root relative squared error            139.686  %
Total Number of Instances             1800
```

It be Training map used WindowsPC signals and testing by Iphone 5s with K=1.

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          52               2.8889 %
Incorrectly Classified Instances      1748              97.1111 %
Kappa statistic                          0.0011
Mean absolute error                      0.054
Root mean squared error                  0.2298
Relative absolute error                 99.9044 %
Root relative squared error            139.8421 %
Total Number of Instances             1800
```

It be Training map used WindowsPC signals and by Macbook with K=1.

```
=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances          66               3.6667 %
Incorrectly Classified Instances      1734              96.3333 %
Kappa statistic                          0.0091
Mean absolute error                      0.0535
Root mean squared error                  0.2302
Relative absolute error                 99.0924 %
Root relative squared error            140.0711 %
Total Number of Instances             1800
```

Test and training with different device results changing k value

Now we choice the map trained by WindowsPc signals and testing by Macbook changing the value of K. We wont to see if changing the K value improves the Kappa value or the error.

It be Training map used WindowsPC signals and testing with the same device Macbook with K=3.

Time taken to build model: 0 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	90	5 %
Incorrectly Classified Instances	1710	95 %
Kappa statistic	0.0229	
Mean absolute error	0.053	
Root mean squared error	0.2248	
Relative absolute error	98.0978 %	
Root relative squared error	136.7986 %	
Total Number of Instances	1800	

It be Training map used WindowsPC signals and testing with the same device Macbook with K=5.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	126	7 %
Incorrectly Classified Instances	1674	93 %
Kappa statistic	0.0434	
Mean absolute error	0.0523	
Root mean squared error	0.2222	
Relative absolute error	96.785 %	
Root relative squared error	135.2266 %	
Total Number of Instances	1800	

It be Training map used WindowsPC signals and testing with the same device Macbook with K=6.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	136	7.5556 %
Incorrectly Classified Instances	1664	92.4444 %
Kappa statistic	0.0491	
Mean absolute error	0.0522	
Root mean squared error	0.2215	

Relative absolute error	96.7216 %	
Root relative squared error	134.7684 %	
Total Number of Instances	1800	

It be Training map used WindowsPC signals and testing with the same device Macbook with K=7.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	140	7.7778 %
Incorrectly Classified Instances	1660	92.2222 %
Kappa statistic	0.0514	
Mean absolute error	0.0523	
Root mean squared error	0.2215	
Relative absolute error	96.8963 %	
Root relative squared error	134.8119 %	
Total Number of Instances	1800	

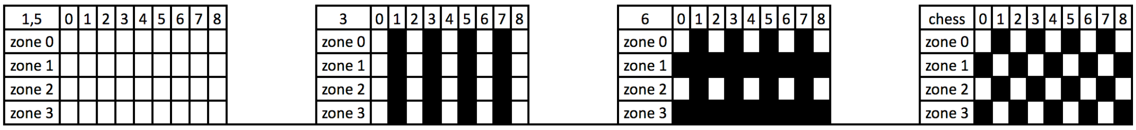
It be Training map used WindowsPC signals and testing with the same device Macbook with K=10.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	120	6.6667 %
Incorrectly Classified Instances	1680	93.3333 %
Kappa statistic	0.04	
Mean absolute error	0.0524	
Root mean squared error	0.2213	
Relative absolute error	97.066 %	
Root relative squared error	134.6845 %	
Total Number of Instances	1800	

Like we see in the last results, the error is quasi the same, that means that K value does not affect and we cold not use it for improve our scenario.

Changing the coverage area on room 130

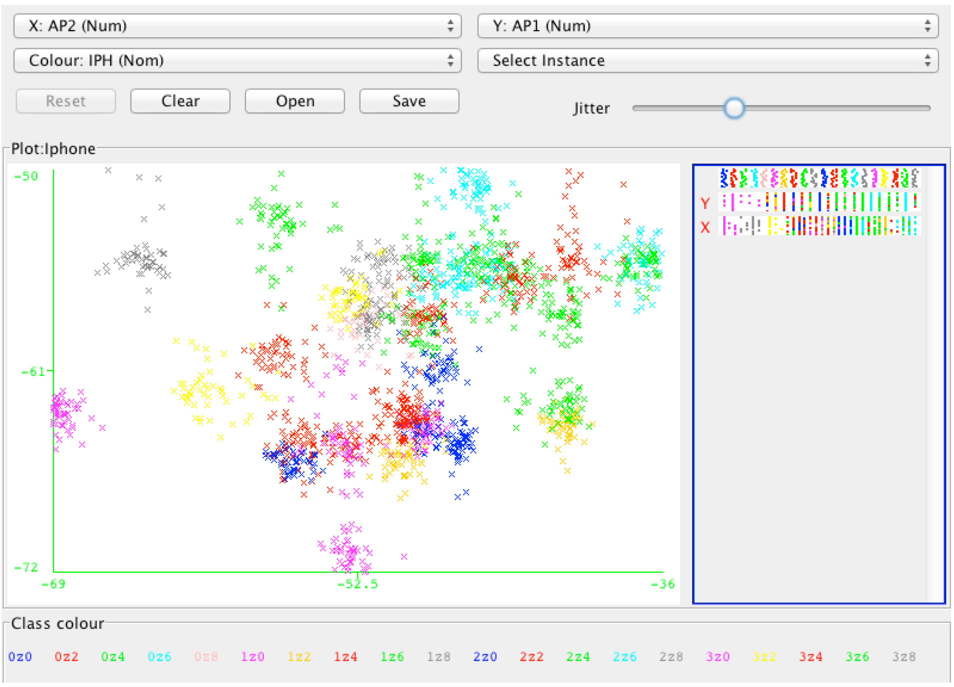


At this point we try to define a new value for the area. Like we show on the before graph using the initial precision areas is not possible determine a concrete point because exist a big concentration of signals in the same zone. For this reason now we will work with different areas starting on $3m^2$, trying to use the dispersion areas were is more easy make localization.

Power Training distribution Iphone 3m2

=== Evaluation on training set ===
=== Summary ===

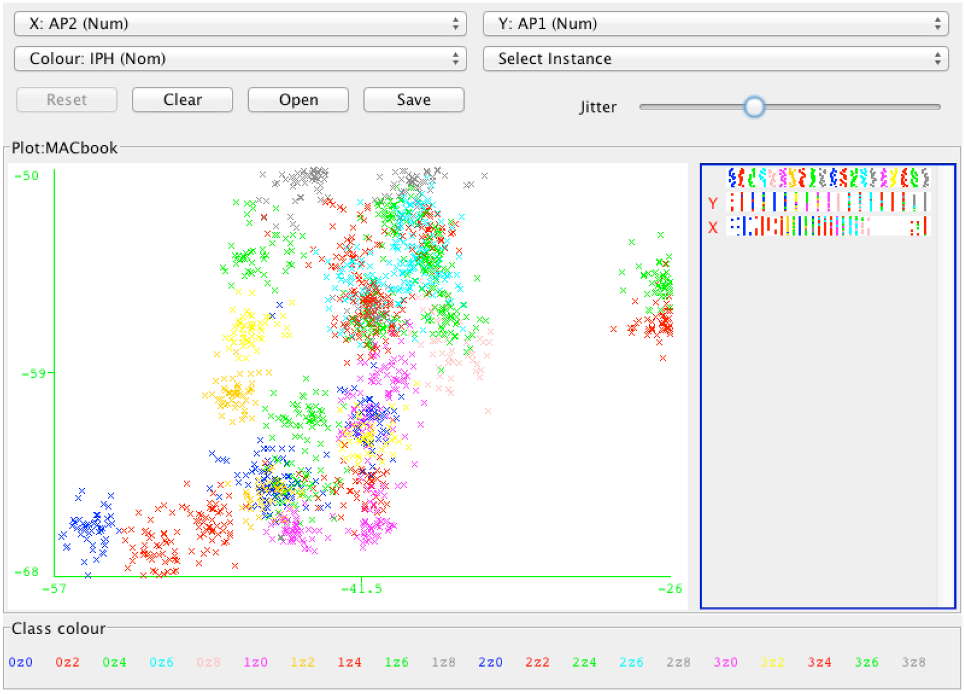
Correctly Classified Instances	837	83.7%
Incorrectly Classified Instances	163	16.3 %
Kappa statistic	0.8284	
Mean absolute error	0.0207	
Root mean squared error	0.1011	
Relative absolute error	21.7791 %	
Root relative squared error	46.406 %	
Total Number of Instances	1000	



Power training distribution Macbook 3m2

=== Evaluation on training set ===
=== Summary ===

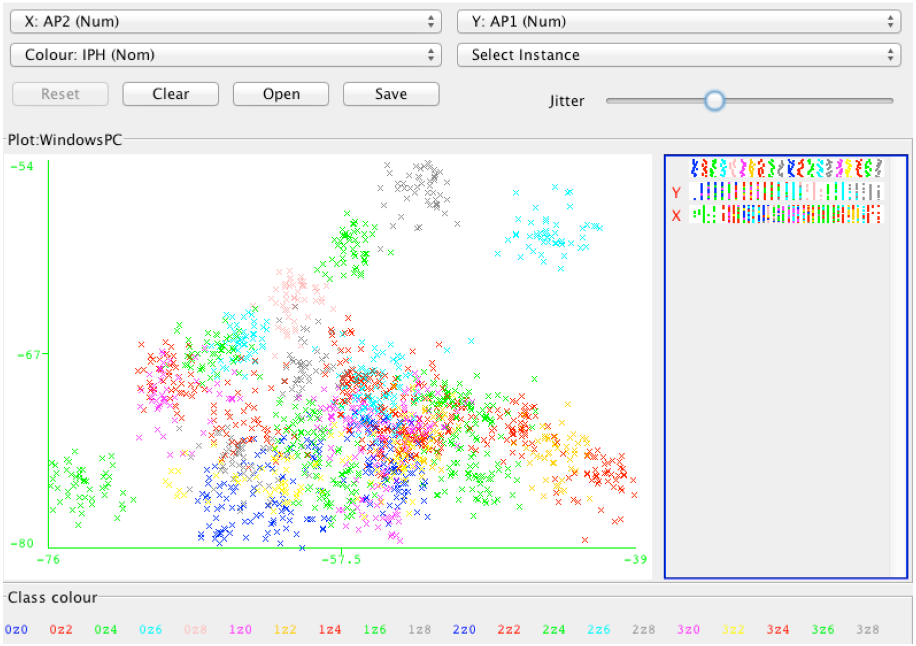
Correctly Classified Instances	820	82	%
Incorrectly Classified Instances	180	18	%
Kappa statistic	0.8105		
Mean absolute error	0.0223		
Root mean squared error	0.1051		
Relative absolute error	23.4642 %		
Root relative squared error	48.2342 %		
Total Number of Instances	1000		



Power training distribution WindowsPC 3m2

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	838	83.8 %
Incorrectly Classified Instances	162	16.2 %
Kappa statistic	0.8295	
Mean absolute error	0.0215	
Root mean squared error	0.1027	
Relative absolute error	22.598 %	
Root relative squared error	47.1028 %	
Total Number of Instances	1000	

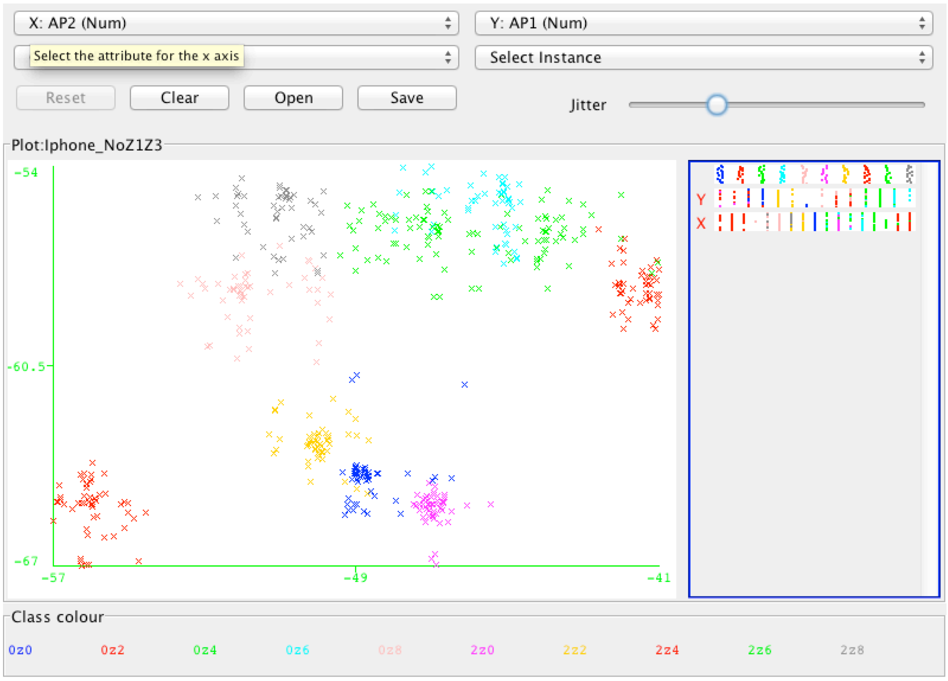


When we increase the area the concentration points be reduced. In this case we are working with an area of 3m2. On the Windows-PC, the cloud points continue been bigger. For this reason it is not possible identify different areas, been this device the most critical. Now we test a area of 6m2.

Power Training distribution Iphone 6m2

=== Evaluation on training set ===
=== Summary ===

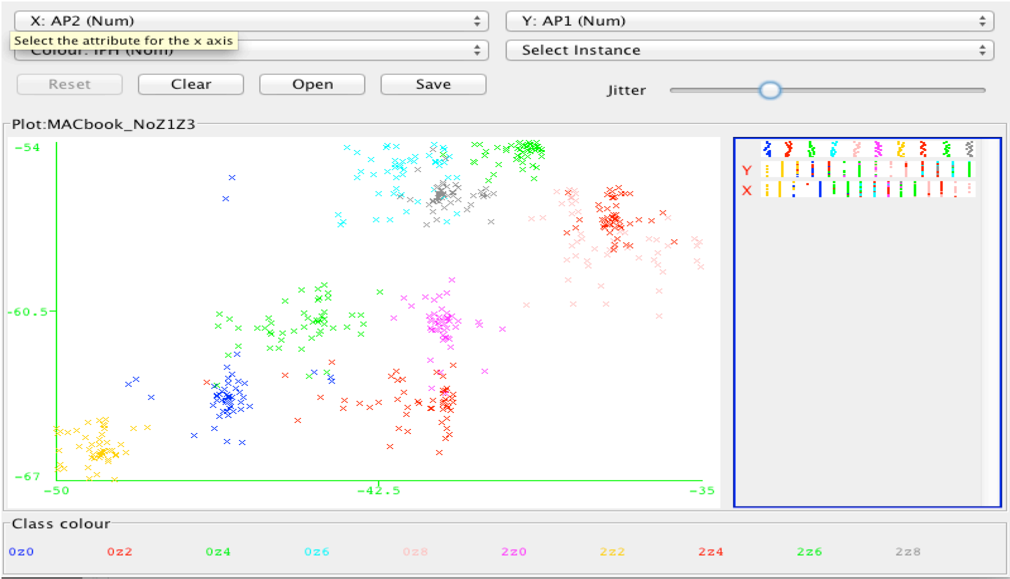
Correctly Classified Instances	477	95.4 %
Incorrectly Classified Instances	23	4.6 %
Kappa statistic	0.9489	
Mean absolute error	0.0144	
Root mean squared error	0.0832	
Relative absolute error	7.9986 %	
Root relative squared error	27.7313 %	
Total Number of Instances	500	



Power training distribution Macbook 6m2

=== Evaluation on training set ===
=== Summary ===

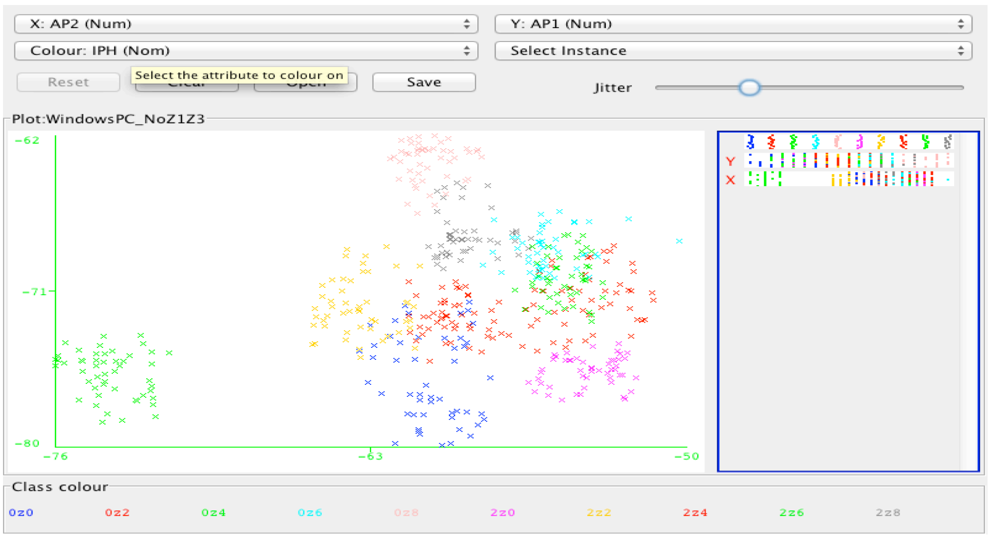
Correctly Classified Instances	473	94.6 %
Incorrectly Classified Instances	27	5.4 %
Kappa statistic	0.94	
Mean absolute error	0.0156	
Root mean squared error	0.0869	
Relative absolute error	8.6778 %	
Root relative squared error	28.9695 %	
Total Number of Instances	500	



Power training distribution WindowsPC 6m2

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	440	88 %
Incorrectly Classified Instances	60	12 %
Kappa statistic	0.8667	
Mean absolute error	0.0304	
Root mean squared error	0.1215	
Relative absolute error	16.8998 %	
Root relative squared error	40.4861 %	
Total Number of Instances	500	



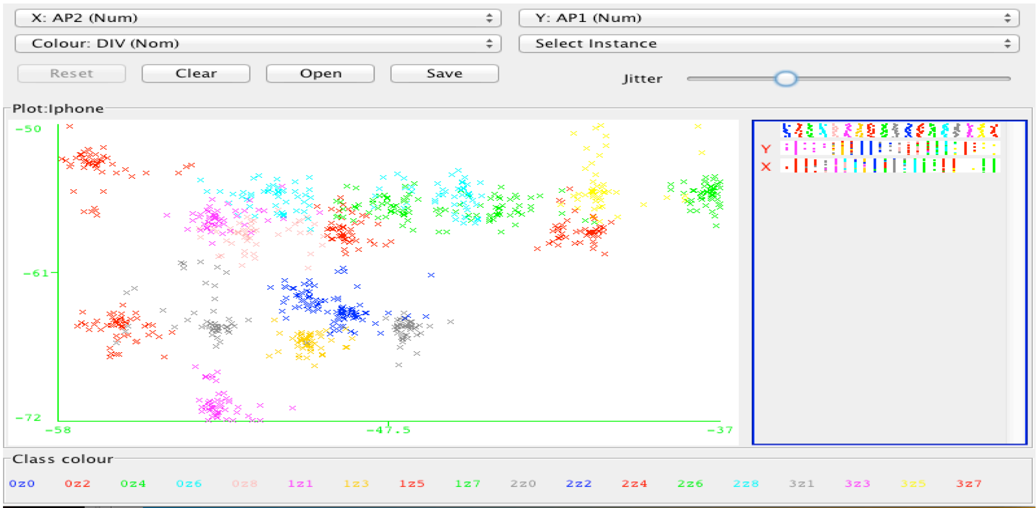
When we increase the precision areas improves the recognition surface. We show this reality on the before pictures for $6m^2$.

At this moment we experiment with other combination. Taking in consideration the mapping area we eliminate the measurements of some areas. To determine which areas we will eliminate we follow a structure like ‘chess’.

Power training distribution IPhone 5S Chess

=== Evaluation on training set ===
=== Summary ===

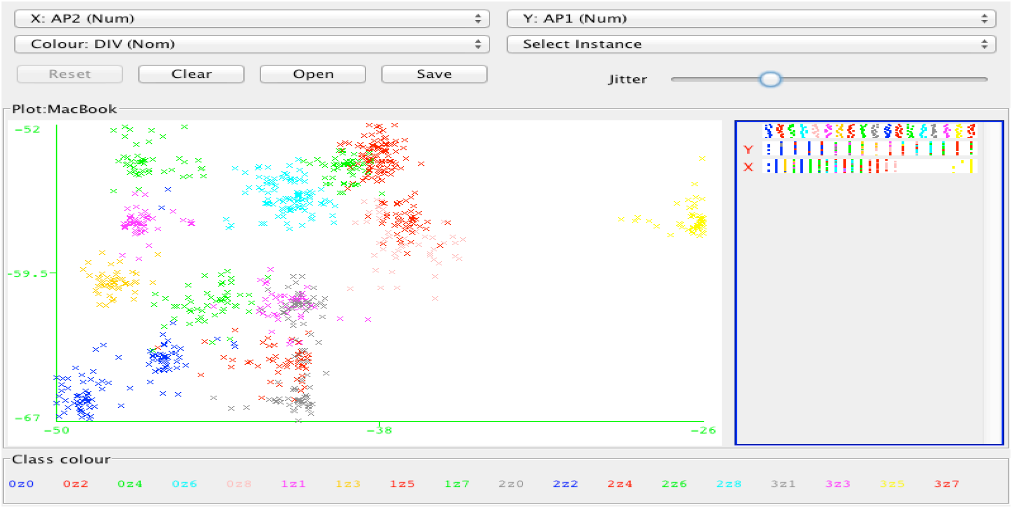
Correctly Classified Instances	845	93.8889 %
Incorrectly Classified Instances	55	6.1111 %
Kappa statistic	0.9353	
Mean absolute error	0.0103	
Root mean squared error	0.0707	
Relative absolute error	9.8177 %	
Root relative squared error	30.8615 %	
Total Number of Instances	900	



Power training distribution MacBook Chess

=== Evaluation on training set ===
=== Summary ===

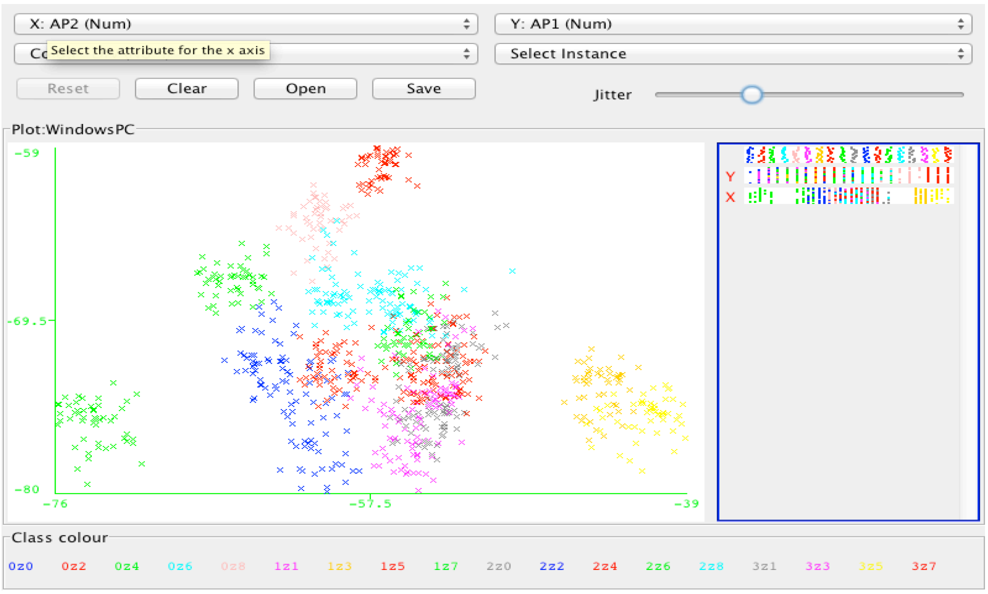
Correctly Classified Instances	821	91.2222 %
Incorrectly Classified Instances	79	8.7778 %
Kappa statistic	0.9071	
Mean absolute error	0.0136	
Root mean squared error	0.0815	
Relative absolute error	12.9208 %	
Root relative squared error	35.6004 %	
Total Number of Instances	900	



Power training distribution Windows-PC Chess

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	753	83.6667 %
Incorrectly Classified Instances	147	16.3333 %
Kappa statistic	0.8271	
Mean absolute error	0.0235	
Root mean squared error	0.1073	
Relative absolute error	22.3745 %	
Root relative squared error	46.8601 %	
Total Number of Instances	900	



Comparative table

	Iphone 5S							
Comparative on training set	**1.5m2**		**3m2**		**6m2**		**Chess**	
Correctly Classified Instances	837	84%	837	84%	477	95%	845	94%
Incorrectly Classified Instances	163	16%	163	16%	23	5%	55	6%
Kappa statistic	0.8284		0.8284		0.9489		0.9353	
Mean absolute error	0.0207		0.0207		0.0144		0.0103	
Root mean squared error	0.1011		0.1011		0.0832		0.0707	
Relative absolute error		22%		22%		8%		10%
Root relative squared error		46%		46%		28%		31%
Total Number of Instances	1000		1000		500		900	

	MacBook Pro							
Comparative on training set	**1.5m2**		**3m2**		**6m2**		**Chess**	
Correctly Classified Instances	1367	76%	820	82%	473	95%	821	91%
Incorrectly Classified Instances	433	24%	180	18%	27	5%	79	9%
Kappa statistic	0.7526		0.8105		0.94		0.9071	
Mean absolute error	0.017		0.0223		0.0156		0.0136	
Root mean squared error	0.092		0.1051		0.0869		0.0815	
Relative absolute error		32%		23%		9%		13%
Root relative squared error		56%		48%		29%		36%
Total Number of Instances	1800		1000		500		900	

	Windows PC							
Comparative on training set	**1.5m2**		**3m2**		**6m2**		**Chess**	
Correctly Classified Instances	837	84%	838	84%	440	88%	753	84%
Incorrectly Classified Instances	163	16%	162	16%	60	12%	147	16%
Kappa statistic	0.8284		0.8295		0.8667		0.8271	
Mean absolute error	0.0207		0.0215		0.0304		0.0235	
Root mean squared error	0.1011		0.1027		0.1215		0.1073	
Relative absolute error		22%		23%		17%		22%
Root relative squared error		46%		47%		40%		47%
Total Number of Instances	1000		1000		500		900	

Using three access point like reference

We introduce other AP to try to determine improved behaviors. We select the AP3 with BSSID E8:04:62:F6:C4:82 and it combine with the AP1 and AP2.

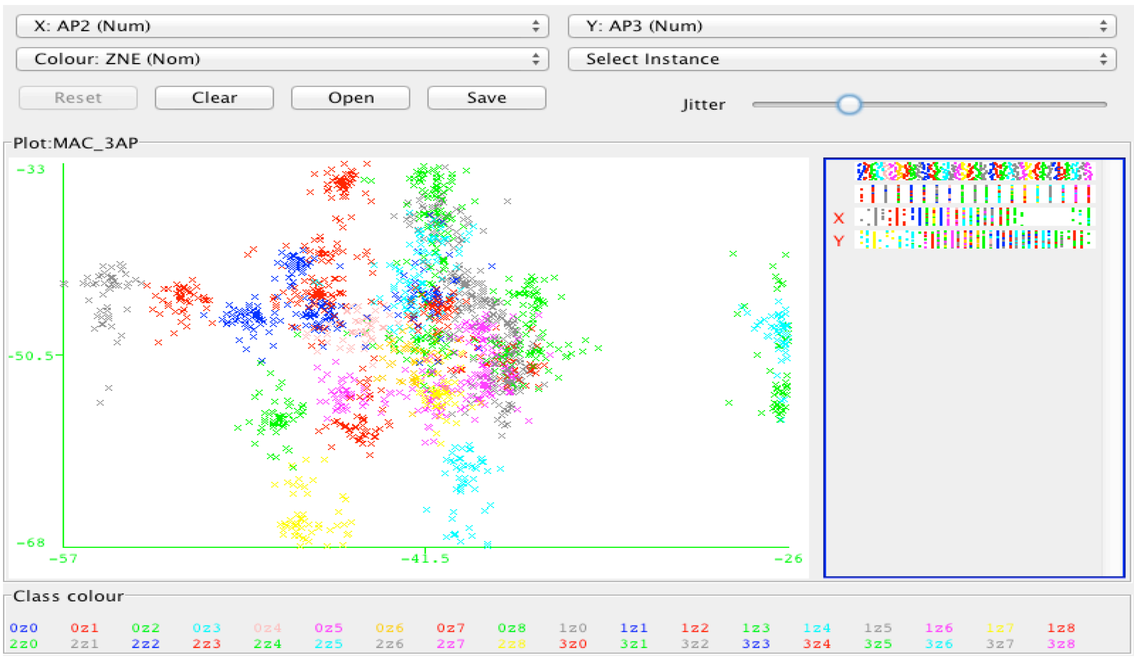
It be Training map used Macbook-pro signals and testing with the same device MacBook with K=1

=== Stratified cross-validation ===
=== Summary ===

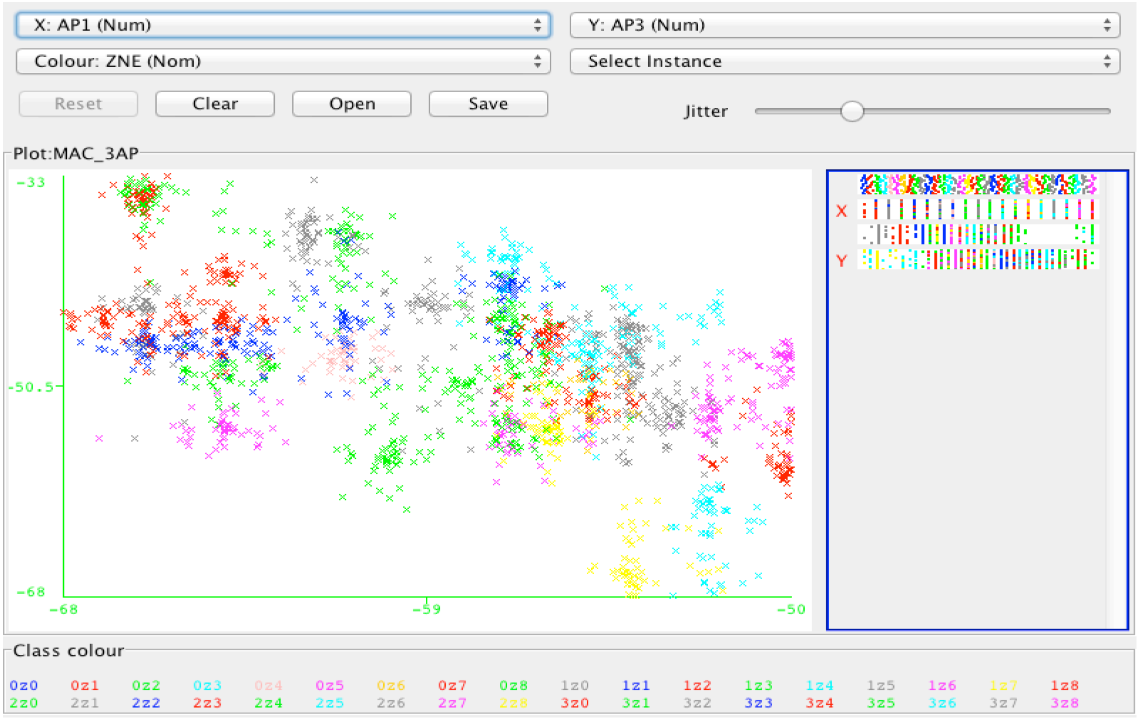
Correctly Classified Instances	1655	91.9444 %
Incorrectly Classified Instances	145	8.0556 %
Kappa statistic	0.9171	
Mean absolute error	0.0054	
Root mean squared error	0.0607	
Relative absolute error	10.0755 %	
Root relative squared error	36.9221 %	
Total Number of Instances	1800	

When we plot the combination signals we show that appears a confuse areas where is difficult define a concrete zone.

Power training MAC AP2 vs AP3



Power training MAC AP1 vs AP3



.

Normalize dates

Taking into account the range of power between the different devices we normalize the dates for the 3 devices and repeat all text The normalization been between (0:-1).

It be Training map used MacBook-pro signals and testing by it self with K=1.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	1367	75.9444 %
Incorrectly Classified Instances	433	24.0556 %
Kappa statistic	0.7526	
Mean absolute error	0.017	
Root mean squared error	0.092	
Relative absolute error	31.5051 %	
Root relative squared error	55.9964 %	
Total Number of Instances	1800	

It be Training map used Windows-PC signals and testing by it self with K=1.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	1352	75.1111 %
Incorrectly Classified Instances	448	24.8889 %
Kappa statistic	0.744	
Mean absolute error	0.0178	
Root mean squared error	0.0938	
Relative absolute error	32.9105 %	
Root relative squared error	57.1018 %	
Total Number of Instances	1800	

It be Training map used Iphone5s signals and testing by it self with k=1.

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances	1450	80.5556 %
Incorrectly Classified Instances	350	19.4444 %
Kappa statistic	0.8	
Mean absolute error	0.0139	
Root mean squared error	0.083	
Relative absolute error	25.6948 %	
Root relative squared error	50.4893 %	
Total Number of Instances	1800	

It be Training map used Iphone5s signals and testing by MacBook with k=1.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	95	5.2778 %
Incorrectly Classified Instances	1705	94.7222 %
Kappa statistic	0.0257	
Mean absolute error	0.0534	
Root mean squared error	0.2179	
Relative absolute error	98.9552 %	
Root relative squared error	132.6092 %	
Total Number of Instances	1800	

It be Training map used Iphone5s signals and testing by Windows PC with k=1

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	30	1.6667 %
Incorrectly Classified Instances	1770	98.3333 %
Kappa statistic	-0.0114	
Mean absolute error	0.0546	
Root mean squared error	0.2238	
Relative absolute error	101.079 %	
Root relative squared error	136.1562 %	
Total Number of Instances	1800	

It be Training map used MacBook-pro signals and testing by IPhone 5s with K=1.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	125	6.9444 %
Incorrectly Classified Instances	1675	93.0556 %
Kappa statistic	0.0429	
Mean absolute error	0.0518	
Root mean squared error	0.222	
Relative absolute error	95.8127 %	
Root relative squared error	135.0917 %	
Total Number of Instances	1800	

It be Training map used MacBook-pro signals and testing by Windows PC with K=1.

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	62	3.4444 %
Incorrectly Classified Instances	1738	96.5556 %
Kappa statistic	0.0069	

Mean absolute error	0.0536
Root mean squared error	0.2227
Relative absolute error	99.1754 %
Root relative squared error	135.5428 %
Total Number of Instances	1800

The behavior of the data normalize is worse than without. The results had not been like we expected. In this case we expected improve the behavior working with the margin instead of the power value.

True positive analysis

		0	1	2	3	4	5	6	7	8
MACbook	Zone 0	0,88	0,1	0,88	0,84	0,94	0,72	0,78	0,66	0,92
	Zone 1	0,94	0,34	0,4	1	0,36	0,86	0,56	0,98	0,96
	Zone 2	0,82	1	0,96	1	0,82	0,76	0,88	0,74	0,18
	Zone 3	1	0,8	0,78	1	0,74	0,96	0,9	0,8	0,96

		0	1	2	3	4	5	6	7	8
WindowsPC	Zone 0	0,8	0,9	0,28	0,92	1	0,88	0,86	0,8	0,96
	Zone 1	0,62	0,72	0,46	1	0,92	0,5	1	0,82	1
	Zone 2	0,88	0,58	0,76	0,32	0,78	1	0,58	1	0,94
	Zone 3	0,76	0,58	0,82	0,48	0,82	0,88	0,66	1	0,64

		0	1	2	3	4	5	6	7	8
Iphone 5S	Zone 0	0,9	0,98	0,86	0,32	0,34	0,6	0,7	0,64	0,36
	Zone 1	0,16	1	0,94	0,92	0,78	0,8	0,92	0,84	0,8
	Zone 2	1	0,94	0,92	0,9	0,94	0,92	0,8	1	0,96
	Zone 3	1	0,86	0,98	0,8	0,96	0,92	0,82	0,98	1

Area parts comparison

Now would focused in determined if the surface of room affect the power distribution or if is possible fined a pattern that repeats for all devices. To do this we had analyzed the devices testing results training.

We show the true positive (TP) results. This is a value between 0-1. When the value is 1 this means 100% success, when it is 0,5 this means 50% success and so on.

		0	1	2	3	4	5	6	7	8
Zone 0	**MACbook**	0,88	0,1	0,88	0,84	0,94	0,72	0,78	0,66	0,92
	WindowsPC	0,8	0,9	0,28	0,92	1	0,88	0,86	0,8	0,96
	IPHONE	0,9	0,98	0,86	0,32	0,34	0,6	0,7	0,64	0,36
Zone 1	**MACbook**	0,94	0,34	0,4	1	0,36	0,86	0,56	0,98	0,96
	WindowsPC	0,62	0,72	0,46	1	0,92	0,5	1	0,82	1
	IPHONE	0,16	1	0,94	0,92	0,78	0,8	0,92	0,84	0,8
Zone 2	**MACbook**	0,82	1	0,96	1	0,82	0,76	0,88	0,74	0,18
	WindowsPC	0,88	0,58	0,76	0,32	0,78	1	0,58	1	0,94
	IPHONE	1	0,94	0,92	0,9	0,94	0,92	0,8	1	0,96
Zone 3	**MACbook**	1	0,8	0,78	1	0,74	0,96	0,9	0,8	0,96
	WindowsPC	0,76	0,58	0,82	0,48	0,82	0,88	0,66	1	0,64
	IPHONE	1	0,86	0,98	0,8	0,96	0,92	0,82	0,98	1

.

APPENDIX E. REDPIN

RedPin [1] consist of two basic components: a Sniffer and a Locator component. While the Sniffer-component has to run on the mobile device for obvious reasons, the Locator-component can be run either on a central server or on each mobile device separately. Although running the Locator, and hence storing the fingerprints, locally would be beneficial considering the users privacy, we need to store this data on a central server in order to allow for users collaborating.

RedPin implements the Locator as a server, using Java and MySQL as illustrated in Figure 29. It uses i.e. Awere-API for all communication aspects, and IOS 7.4 for the Sniffer component. This separation was necessary, as only the IOS API would allow us to get the information we wanted to collect. For both the server and the mobile client, handle the serialization, transmission, and storage of the measured data. Communication between the server and the client is done through asynchronous JSON messaging using a polling mechanism.

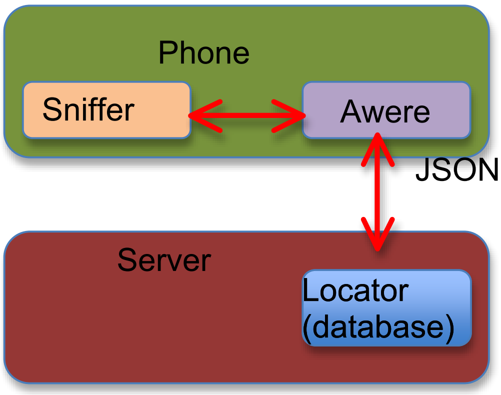


Figure 29. RedPin architecture

Server

The RedPin server provides several services for mobile clients. First of all, it provides a service that allows it to store fingerprints in a central database. This service is called whenever a mobile user stores or redefines a location. Another service allows the mobile clients to retrieve maps, i.e., images of the floor plan that are associated with a certain location. And most importantly, the server provides a service to locate a mobile device, i.e., to retrieve the fingerprint, and thus the location that best matches the measurement taken by the mobile client.

Positioning

Because a location is simply expressed by a symbolic identifier in RedPin, the problem of calculating the current position is reduced to the problem of finding one fingerprint that best matches the given measurement. Hence, to determine the current location of a mobile device, we need to find the one known fingerprint that matches the current measurement best.

Sniffer Measurements

RedPin measures three different signal sources, namely GSM, Wi-Fi, and Bluetooth. A wide range of signal strength fingerprints have been shown to increase accuracy of indoors localization systems significantly. While both GSM and Wi-Fi signals may fluctuate, Bluetooth devices are not always detected in the very short time range when we scan for devices. As a result, measurements can differ considerably, even when taken at the same place and in short succession. Hence, the biggest advantage of having combined fingerprints of GSM, Wi-Fi, and Bluetooth signals is that we can adapt the localization algorithm dependent on the actual measurement.

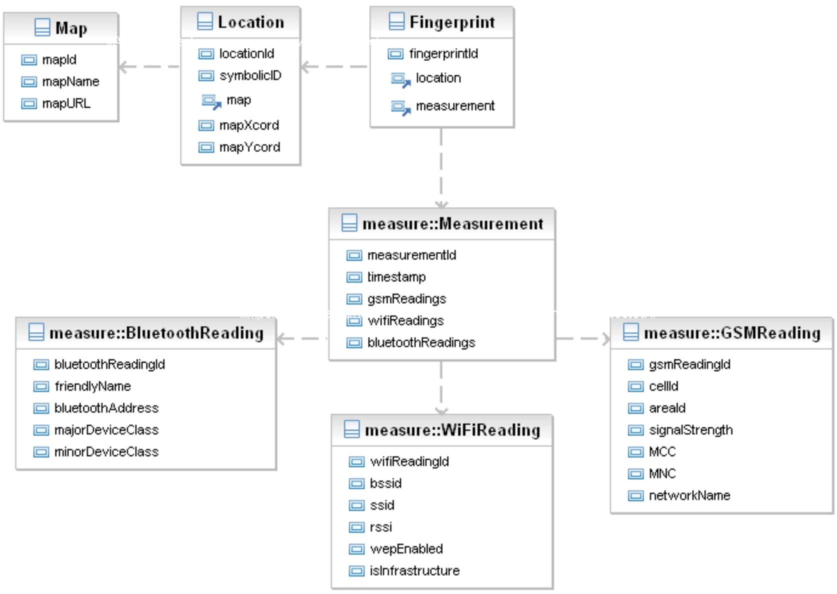


Figure 30. RedPin data model[3]

To create an internationally unique GSM identifier, RedPin reads out the cell identifier (CI), the mobile country code (MCC), the mobile network code (MNC), as well as the location area code (LAC). In addition, we also retrieve the current received signal strengths (RSS) as an absolute value.

Bluetooth devices can be uniquely identified by the Bluetooth device address (BD ADDR), similar to the MAC addresses of a network card. However, as we

3 RedPin - Adaptive, Zero-Configuration Indoor Localization through User Collaboration (Philipp Bolliger)

only want to consider non-portable devices, we have to retrieve the major and the minor device class during inquiry. This way, we can ignore mobile devices like mobile phones or portable audio devices that would distort the result otherwise.

```
measurement = '{' id ['"timestamp":' timestamp ','] ['"gsmReadings":' gsm ',']
 ['"bluetoothReadings":' bluetooth ',']'"wifiReadings":' wifi '}'

wifi = '[' [ wifireading {',' wifireading } ] ']'
wifireading = '{' id '"bssid":' String ',' '"ssid":' String ','
            '"rssi":' Integer ',' '"wepEnabled":' bool ','
            '"isInfrastructure":' bool '}'

gsm = '[' [ gsmreading {',' gsmreading } ] ']'
gsmreading = '{' id '"cellId":' String ',' '"areaId":' String ','
            '"signalStrength":' String ',' '"MCC":' String ','
            '"MNC":' String ',' '"networkName":' String '}'

bluetooth = '[' [ bluetoothreading {',' bluetoothreading } ] ']'
bluetoothreading = '{' id '"friendlyName":' String, ','
            '"bluetoothAddress":' String ',' '"majorDeviceClass":' String ','
            '"minorDeviceClass":' String '}'

action = '"setFingerprint"' | '"getLocation"' | '"getMapList"'| '"setMap"' | '"removeMap"' | '"getLocationList"'
| '"updateLocation"' | '"removeLocation"'
```

Figure 31. Syntax Code Signal

The *Figure 31* shows the syntax code to identify the different signals related with the mapping-server. All the objects are easy to understand only make an observation related with the Fingerprint. When in the area of mapping-map has defined the corresponded signals, over this, it generates a fingerprint. These will be very useful to establish the communication between the client and the server.

Locator Algorithm

To locate a mobile device, the Locator compares the current measurement with all known fingerprints in the database by calculating the distance measure as described above. If a fingerprint can be found whose distance to the current measurement is smaller than the threshold, the associated location will be returned to the mobile device. If multiple fingerprints are found, the system will return the best match.

.